AWS Well-Architected

Change Enablement in the Cloud

Change Enablement in the Cloud: AWS Well-Architected

Table of Contents

Change Enablement in the Cloud

Publication date: **June 27, 2024** (*Document revisions*)

Like every business function, change management should provide the capability for your organization to succeed. Just as every business has a finance function to govern spending, change management is essential for optimizing business risk. You can benefit from change management if you have migrated to the cloud, have a hybrid environment, or architected your workload natively in the cloud. An effective change management process is agile and reduces time to market while optimizing risk to your business. With an efficient process, you can improve the quality and speed of delivery of all changes for the business. An effective record of change should also act as one of your first troubleshooting references if an incident occurs.

Typically, businesses transitioning to the cloud expect to iterate on products and deliver enhanced business value through an increased volume of changes to their IT infrastructure and applications. Current industry trends indicate that a high performing change management process is comprised of increased frequency of deployment, improved lead time for a change, and lower failure rates. However, to achieve this, it also requires version control and continuous delivery, as well as automated testing and deployment capabilities. Research also suggests that peer-reviewed changes within a team can achieve the same level of risk optimization as using a change advisory board. This is important because it allows organizations to decentralize change approval authority. (ITIL® 4: High-Velocity IT, PeopleCert+)

By following the guidance in this paper, updating your processes aligns with both the AWS Cloud Adoption Framework (CAF) and the AWS Well-Architected Operational Excellence Framework, ensuring adherence to best practices in governance when deploying changes to your AWS environment.

Introduction

In a cloud computing environment, new IT resources are readily available. This reduces the time it takes to make resources available to your developers, which leads to a dramatic increase in agility for the organization. As a result, the cost and time it takes to experiment and develop is significantly lower. For more detail, see Six advantages of cloud computing.

The more successful an organization is at increasing its agility in the cloud, the more difficult it can become to manage change. Traditional processes can impede the volume and speed the

cloud enables. Business stakeholders may have become accustomed to longer release cycles using waterfall methodologies, and challenges can arise from the transition to working in a way that increases the frequency of software releases. These challenges may result in increased stakeholder engagement, the introduction of unnecessary gates that hinder development progress, or unmanaged changes. Modernizing your change management process will be necessary to overcome these challenges.

Make frequent, small, and reversible changes. Design workloads to allow smaller components to be updated more frequently. Plan your changes in smaller increments that can be reversed if they fail (without an impact to business outcomes). Use this approach to achieve the desired agility. AWS best practices and strategies for designing and operating a cloud workload align with this approach. A change process must continue to govern the deployment of a new services, software, patches, and configuration changes.

In the cloud, you can enable automated governance with a complete audit trail of the deployment steps. Due to automated deployment capabilities, you can preserve agility by reducing the penalty if a rollback of a failed change is required. To achieve cloud agility, organizations must rollback changes that have adverse business consequences, and automate this process where possible. The automation should enforce change policies to occur within the software development pipeline as regularly scheduled and unscheduled changes continually flow. Different policies and procedures should exist for emergency changes or changes that require manual processes during deployment.

What is ITIL?

The framework operated by PeopleCert+ defines an internationally recognized approach to IT service management (ITSM). ITIL enhances ISO 20000, which provides a formal and universal standard for organizations seeking to have their service management capabilities audited and certified. ITIL®4 goes one step further than previous versions to define IT practices necessary to achieve the standard through use of the latest industry methodologies of DevOps, LEAN, and agile. ITIL®4 provides a solid foundation for defining and maturing cloud capabilities. ITIL and the AWS Cloud Adoption Framework (CAF) are compatible. Like ITIL, CAF organizes and describes the activities and processes involved in planning, creating, managing, and supporting modern IT services. The operations perspective in CAF offers practical guidance and comprehensive guidelines for establishing, developing, and operating your cloud-based IT capabilities.

In response to an adoption of modern development technologies and techniques for high velocity IT, ITIL®4 has renamed change management to *change enablement*. This broadens the scope of change management to focus less on control and more on the delivery of business outcomes by addressing the impacts of a change, people, and processes, with a focus on velocity. The purpose of a change enablement practice remains much the same as it was in previous versions of ITIL: to provide best practices for organizations to deliver new services with speed and quality while properly managing and optimizing business risks. This new version embraces a recognition for automation and decentralization of authority to approve changes, especially as part of a DevOps lifecycle.

As defined by PeopleCert+, ITIL comprises five volumes and a guide document for each practice. The Practice Guides are additional documents that give more detailed guidance on implementing each of the practices. The guide documents are updated more frequently with the latest best practices.

ITIL®4 volume	Description
High-Velocity IT	Focuses on digital products, digital transform ation, and IT services, including customer experiences, and good organizational practices as business and technology converge.
Create, Deliver, & Support	Provides practical guidance in applying the concepts described in ITIL to create, deliver,

ITIL®4 volume	Description
	and support IT services. It demonstrates how to integrate service management practices into end-to-end value streams.
Direct, Plan, & Improve	Helps to align product and service management disciplines with modern business trends needed to drive successful organizat ional transformation through continual improvement.
Drive Stakeholder Value	Provides guidance on establishing, maintaini ng, and developing effective service relations hips at appropriate levels.
Digital & IT Strategy	Provides guidance on establishing IT and digital strategies in alignment with corporate strategy. It directs IT leaders to understand how the organization's strategy impacts the design, delivery, and support of IT services. They target this guidance towards C-suite professionals who lead business and IT organizations.
Individual Practice Guide Documents	ITIL®4 has provided individually written papers that provide guidance to define and operate the ITIL®4 practices. "The Change Enablemen t ITIL®4 Practice Guide" provides details on the latest change management practices. Each ITIL practice has its own practice guide for reference. These can be accessed in the PeopleCert+ website after one receives the appropriate ITIL certification.

Change enablement in ITIL®4

ITIL®4 defines a service change as "the addition, modification, or removal of anything that could have a direct or indirect effect on services (ITIL®4 Change Enablement Practice Guide, PeopleCert +)."

The purpose of the change enablement practice is to maximize the number of successful service and product changes by properly assessing risks, authorizing changes to proceed, and managing the change schedule (ITIL®4 Change Enablement Practice Guide, PeopleCert+). This enables a DevOps culture.

The goal is to increase velocity while maintaining the same or improved ability to mitigate risks. This requires part of the IT value stream to be automated with CI/CD capabilities. The mentality of *you build it, you run it*, driven by DevOps, forces IT organizations to rethink how they deploy changes. The team that builds a service or an application must also support it, instead of simply handing it to another team for deployment and support. Application development teams are now becoming fully accountable for both the operation of the service they build and the quality of code they deploy. In the cloud, this includes infrastructure as code (IaC).

ITIL®4 now considers the impact of changes to a business value stream with higher importance than previous versions. The benefits of this improved approach to change management foster both risk management and velocity, encouraging the organization to define their change management process so that they cannot have one without the other.

Change enablement in the cloud

All changes should be delivering business value. The change enablement practice should focus on optimizing business risk, maximizing productivity, and minimizing wasted effort and cost. The AWS Cloud enhances a change enablement practice by:

- Minimizing the possibility of human error with workflow automation and integration
- Allowing the creation of identical environments for improved predictable and testable outcomes
- Removing the requirement to submit changes to scale infrastructure to meet business demand
- Automatically recovering from failure and rolling back failed changes
- Blue/green and other deployment techniques

Automation reduces the business risk associated with a change and increases business agility, which delivers more business value. Ultimately, this is why we make changes to applications and infrastructure. Agile and DevOps ways of working are also designed to deliver business value more quickly. However, to achieve these outcomes, key areas of your change enablement process may need to be updated.

AWS supports the evolution of your process. AWS Systems Manager Change Manager is an AWS product for requesting, approving, implementing, and reporting on operational changes to your application and infrastructure. If you use AWS Organizations, you can manage changes from a single delegated administrator account across multiple AWS accounts and across AWS Regions. Alternatively, using a local account, you can manage changes within a VPC. Use Change Manager for managing changes to both AWS resources and on-premises resources.

In an AWS architecture, you can isolate applications within a single VPC, which helps you reduce the scope when making changes and decentralize the authority for change approval. This isolation is not possible with on-premises infrastructure solutions. Decentralizing change authority through enforced automation increases velocity as desired. Customers should follow Well-Architected Best Practices when architecting their workloads to optimize operational capabilities.

Managing configuration items in the cloud

On AWS, you can manage cloud configuration items in AWS Config when processing changes more efficiently and accurately. For example, if an application suffers a fault in a traditional IT environment where application updates and operating system patches are installed or deployed

on a server, an engineer may be tasked to investigate and either apply a fix or deploy a new server. Either of these tasks requires an emergency change and could impact the business for an amount of time.

The cloud enables automation that is initiated by a set of configured conditions. For example, AWS Auto Scaling groups (ASG) can create additional EC2 instances as needed. You can use an ASG to automate the update of system patches by allowing the ASG to deploy each patched EC2. In this example, the manageable resource, the ASG, is recognized as the configuration item, not the EC2 instances that are created. Furthermore, if you use a health check to detect a failure within an ASG, the ASG replaces affected items automatically from the designated image. This represents how cloud technology removes the requirement to have change control under certain pre-defined conditions. The automation provisions additional resources to meet demand, reduces human error and configuration drift, and minimizes business risk while reducing the time to recover.

In a traditional environment, the addition of servers requires approval using a standard or normal change record. In the best-case scenario, work was done to increase capacity. In the worst-case scenario, the business was impacted and put at risk by the business processes required to introduce additional capacity, and it may not have been possible to meet the business demand in the required amount of time.

According to the traditional definition of a change (the addition, modification, replacement, or removal of a configuration item), a change record would be necessary when an ASG deploys an EC2 instance. As a result, it may help to redefine what items should be considered configuration items.

As a rule of thumb in the cloud, any resource that is auto-deployed without human triggering, should not be subject to a change ticket, nor should it be considered a configuration item that requires you to manage change. In the previous example, the servers themselves are not configuration items when they are in an Auto Scaling group because they are transient and initiated by a non-human process. The Auto Scaling group and the image used to create the servers should be considered configuration items. When changes are made to these items and configurations are incorrect, the business can be exposed to risk.

To manage configuration items in the AWS Cloud, use AWS Config to assess, audit and evaluate the configuration of AWS resources allowing you to monitor and record AWS resource configurations. With AWS Config, you can track the relationships between resources and review resource dependencies prior to making changes. Once a change occurs, you can quickly review the history of the resource's configuration and determine what the resource's configuration looked like at any point in the past. AWS Config provides you with information to assess how a change to a resource configuration affects your other resources, which minimizes the impact of change-related

incidents. If you still maintain a CMDB within your ITSM platform, you can also use the AWS Service Management Connector to synchronize configuration items between AWS Config and your CMDB.

You can use AWS CloudFormation change sets to preview how proposed changes to a stack might affect your running resources. For example, you can use change sets to check whether your changes would delete or replace any critical resources. When you decide to deploy the change set, AWS CloudFormation automatically makes the changes.

Automation of change models

Regardless of whether a deployment is an application, a patch, or a configuration change, an optimized cloud configuration can automate the deployment process through an unchanged pipeline. You can use cloud and software development pipelines to model change more specifically by defining repetitive end-to-end change types. These change models help you pre-plan for the level of business risk when you deploy a specific change.

ITIL®4 Change Enablement Practice Guide, PeopleCert+

A Change Model is a repeatable approach to the management of a particular type of change.

The concept of a standard change can be confused with a change model. Standard changes are typically defined generically as low risk and well-documented. A change model is defined specifically and is managed the same way each time it is deployed, often via a development pipeline. Change models may have some level of risk, but these risks are mitigated through automated governance.

Pipeline deployment capabilities foster repeatability and consistency across multiple environments and types of changes, as well as automation of software testing, compliance testing, security testing, and functional testing. With cloud technologies, you can programmatically model each type of change through its entire pipeline workflow and enforce change control policies. With this capability, there is no longer a need to model generic changes based on the level of risk. You can now model specific changes based on how their pipeline has mitigated risk. Once certified, the pipeline is free to deploy these changes on-demand because the automation enforces internal approvals. Although this does not verify that no adverse impacts happen, it optimizes your risk by embedding change policies into the pipeline workflow. A change cannot be automatically deployed unless the workflow completes all required steps. This includes the automated creation and population of a change record in your ITSM system.

For example, if an automated security test is approved for deployment, the security review during the change approval process can be reduced or even removed in the appropriate circumstances. Repeatability and consistency throughout the lifecycle of a workload and its deployment reduces the potential time delay and the burden required by the examination of changes by the Change Approval Board. The focus should be on how changes are delivered (through the pipeline) and the automation of tests that reduce required scrutiny by the approval board, both of which are prone to human error.

AWS CodePipeline and Amazon CodeCatalyst help you to automate your software release process and speed up the release of new features to your users. AWS CodePipeline helps you iterate on feedback more quickly. When you automate your build, test, and release process, you can test each code change and catch bugs while they are small and simple to fix. Amazon CodeCatalyst helps you set up a new project from pre-configured blueprints. These two services work together to help you increase velocity.

EC2 Image Builder reduces the effort of keeping images up-to-date and secure by providing a simple graphical interface, built-in automation, and AWS-provided security settings. With Image Builder, there are no manual steps for updating an image, nor do you have to build your own automation pipeline. Creating a golden image using EC2 Image Builder reduces the risk of non-compliant images being used, as well as improving security, consistency, and compliance. For more information, see EC2 Image Builder.

Use AWS Systems Manager to automate operational tasks to help make your teams efficient. With automated approval workflows and runbooks with rich text descriptions, you reduce human error and simplify maintenance and deployment tasks on AWS resources. You can use predefined automation runbooks or build your own to share for common operational tasks, such as stopping and restarting an EC2 instance. Systems Manager also has built-in safety controls that roll out new changes and automatically halt and rollback the change if errors occur. For more information, see AWS Systems Manager.

AWS Systems Manager Change Manager, a subset of the Systems Manager product, provides organizations the ability to automate repeatable operational changes to applications and infrastructure. You can use Change Manager to create automated runbooks using CloudFormation code.

We recommend a change model certification process to ensure that an entire end-to-end pipeline workflow adheres to the requirements of your change control policy and mitigates business risk, while delivering desired business value. The change enablement practice should define clear

criteria necessary for certification of each pipeline that desires to have automated continuous deployment permission.

It is worth noting that from an ITIL perspective, when automating a change model, you are integrating three practices into one value stream: release management, change enablement, and deployment management.

Automated testing and rollback

A change model that has automated testing and rollback creates the confidence for continuous deployment. Enable these capabilities for workloads that require automated change models. Without automated testing and failback capabilities, an automated change model may require you to manually pause the pipeline to gain approval for deployment. This may be acceptable for some workloads, and it highlights another benefit of using the cloud to isolate application workloads. By isolating workloads inside their own account or VPC, we can reduce the overall scope of a change, allowing the workload owner to approve it, even if the changes cannot be automatically deployed.

Remediation from failure

An automated change model that uses deployment capabilities, such as blue/green, should still not be approved without considering the consequences of a failure. A backout plan should be a requirement in the change model. In the event of a failure, automated backout can be initiated manually or automatically based on pre-conditions and status monitoring. Although not always possible with every type of change, the requirement should be that changes are reversible where possible. The application's architecture influences the ability to make changes reversible. Not every change can be easily reversed, but this should be the goal to mitigate any business impact in the case of failure.

As discussed, deployments in the AWS Cloud that use an automated pipeline allow changes to be redeployed quickly and safely, helping minimize risk and reduce business impact. In certain scenarios, it may not be possible to backout changes or redeploy, in which case the organization may need to invoke an incident management process or a business continuity plan to resolve the failed deployment in production. For more detail, see Ensuring Rollback Safety During Deployments. Track metrics for the number of successful vs. reversed changes, not to place blame on individuals, but to reveal continuous improvement opportunities.

For your most critical applications, use continuous data protection in the cloud to provide sub-second recovery point objectives (RPOs) and recovery time objectives (RTOs) in minutes. For more

detail, see AWS Elastic Disaster Recovery. Crucially, where it's not possible to back out changes, the AWS Cloud provides methods to reduce business risk and impact of a failed change by simplifying the redeployment or initiation of disaster recovery plans.

Modern deployment methods in the cloud allow for fast or instant rollback. For example, with blue/green deployments, you can make a change to a workload by deploying an identical copy (green) of the live environment (blue) with the configuration change. Users are then switched to the new environment (green) while the old live environment (blue) remains available, but idle. For more information, see Blue/Green deployments. In this scenario, if a failure is discovered, users can be redirected back to the blue environment, and the business impact can be reduced. It is also possible to combine this approach with a canary release method. With this approach, you can redirect a subset of users to the new deployment, assess its efficacy, and gradually increase the number of users on the new deployment until all users are using the new deployment.

There are other considerations when choosing a method of deployment, but the key is to use automated methods in your pipeline to enforce requirements.

Amazon CodeCatalyst is a unified software development service for development teams to quickly build, deliver and scale applications on AWS while adhering to organization-specific best practices. Developers can automate development tasks and innovate faster with generative AI capabilities, and spend less time setting up project tools, managing CI/CD pipelines, provisioning and configuring various development environments or coordinating with team members. IT Leaders can codify organizational best practices at scale via application blueprints to ensure compliance across teams with scale. For more information, see Amazon CodeCatalyst.

AWS CloudFormation monitors the state of your application during stack creation and updating and rollbacks that operation if the application breaches the threshold of the alarms you've specified. For each rollback you create, you specify the CloudWatch alarm that AWS CloudFormation monitors. AWS CloudFormation monitors the specified alarms during the stack create or update operation and for the specified amount of time after it deploys the resources. If any of the alarms is set off during the stack operation or the monitoring period, AWS CloudFormation rollbacks the entire stack operation. For more information, see AWS CloudFormation.

AWS AppConfig supports best practices by rolling out configuration changes at once or over a period. It monitors the change over a time that customers define. If you configure alarms in Amazon CloudWatch, AWS AppConfig can automatically rollback configuration changes if those alarms are initiated. For more information, see AWS AppConfig.

Adapting your change enablement practice to the cloud

The AWS Cloud facilitates several main adaptations:

1. The ability to automate deployment with deployment techniques that mitigate risk
2. The ability to automate rollback when those deployments fail
3. The automation of change ticket creation from data available in the pipeline, allowing developers to perform all the required steps in their pipeline tool

Because the risk and impact to the business of a failed change can be reduced by these aspects, you can make frequent changes with confidence in the deployment and rollback plans. As a result, your process may also require an update for the acceptance of rolling back changes.

If failed changes have a lower impact due to the speed and consistency of roll back, activating rollbacks is now part of the normal process. This is true when you remediate the issue and push it through the same automated pipelines to deliver the original intended business value of the change.

With these considerations in mind, if automation, pipelines, and deployment methods are in place, it is now possible to reconsider the approach to standard changes. A standard change is where there is a defined event to initiate the change request. In addition, in a standard change, actions are documented and proven, authority is given in advance (or pre-authorized), and the risk is usually low. If the automation, testing, and deployment strategies are put in place, it results in a scenario where large, infrequent, and risky changes are transformed in to small, frequent, and low-risk changes. In this scenario, each standard change evolves to its individual type or model that is specific to its deployable component in production.

By understanding the risk-reduction strategies in the AWS Cloud, and by re-architecting workloads to isolate adjacent workloads and other infrastructure resources, it is possible, and it may even be necessary to widen the scope of a standard change to include deployments that would have previously been considered normal due to their associated risks associated in traditional IT environments. You increase volume and velocity of change while reducing additional risk.

As changes become more frequent with agile methodologies and automation, there is a risk that the process becomes overburdened with normal changes. Higher velocity can lead to delaying changes due to bandwidth or resource constraints, causing important details to be missed. Both scenarios introduce business risk that change enablement aims to optimize. In an environment of small, frequent changes, standard (automated) changes become the new standard. You should

then give proper scrutiny to normal changes, which helps you reduce business risk and deliver on desired business outcomes.

Smaller changes also enable increase in frequency. By changing frequently, you improve your organization's capability, which minimizes business disruption (ITIL® 4: High-Velocity IT, PeopleCert +). This alone can positively impact business value and operational metrics expected from your move to AWS.

Transition of changes

After approval of a release using the change enablement process and following the appropriate project management, release, and deployment management steps, deploying the release causes you to validate and test your change. Guidance on transitioning changes is also provided in the Operational Excellence Pillar of the AWS Well-Architected Framework.

In this stage, you should determine the scope of service validation and testing within the AWS Cloud. This is best illustrated by understanding the AWS Shared Responsibility Model for security. The validation and testing of a service should be limited to the areas in this diagram that are in scope for the customer. However, it is critical that operations have an operational understanding of any managed services before acceptance into service. In addition, AWS informs customers of a change that is being made in their area of responsibility. However, due to the abstraction from the underlying hardware in the cloud, these changes typically do not impact customer business operations.

Automation, integration, and deployment tools in the AWS Cloud allow the business to make small, frequent changes that reduce business risk and introduce business value at an increased rate. The introduction of the cloud should not change the process of service validation and testing, but the rate of change leads to an increased capacity for validation and testing that may require changes to the implementation of the process and the focus of the stakeholders.

Changes introduce business value. It is important that each release meets customer expectations and that IT operations teams can support this new added business value. The criteria for assessing this value in the cloud should not change from what already exists in your legacy process, but the organization must be prepared for an increase in release volume and adapt the implementation of these processes by introducing automation.

A new service requires consent from the customer that the new service meets agreed service-level requirements. The current best practices of tracking your current service-level objectives (SLOs) and tracking service level agreement (SLA) breaches still apply. This can be done by a third-party monitoring service for external facing services. For internal services, you must track this with monitoring and metrics on the primary business function of the services. For example, if a certain business transaction is not performing per the SLA, the monitoring of this performance metric should notify the workload owner of the breach proactively. If this occurs because of a recent change, it's possible to reverse the change with automation. Separate service-level requirements may exist for different aspects of a service, and additional dimensions may be required as metrics

to indicate which aspect is being measured. Indeed, during a change deployment, monitoring can drive an automated rollback if a change would violate an SLA.

Operations must be able to support a new release or service before they make it available to the customer. With the correct tools and process defined, you can design automation that creates documentation, provision runbooks, and build predefined patching plans. This process can be made robust by using cloud services that ensure that only use pre-approved services can be deployed. An example of this is Service Catalog.

The focus of a test manager should be to automate service acceptance testing as much as possible. The cloud can make it easier with a wide variety of tools that are available for both validation and testing.

Amazon CloudWatch provides you with data and actionable insights to monitor your applications running on AWS or on-premises, respond to system-wide performance changes, and display a unified view of operational health. As you deploy changes, you can set alarms, visualize logs and metrics side-by-side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly. For more information, see Amazon CloudWatch.

CloudWatch provides different features, including dashboards, synthetic monitoring, CloudWatch Application Insights, and X-Ray, which can be used during and after transitioning a release into production to ensure that actionable alarms are present to prevent or remediate against service degradation or failure.

Providing easy access to metrics, logs, and dashboards to monitor the health of an application helps teams to resolve problem faster and reduce business risk, while implementing changes.

Reliability

Change implementation can have a direct impact on the availability of workloads and the ability to recover from major incidents or disasters. Change automation is foremost in maximizing application availability. If you have any manual processes, you lose critical time awaiting those manual actions. Theoretically, the smaller in size a change is, the lower the potential impact of that change on the business.

Use deployment patterns that reduce risk, such as blue-green or canary deployments. Perform comprehensive testing in pipelines, including load, performance under load, and resiliency testing. Effective monitoring of the key performance indicators (KPIs) is a requirement, and automated rollback should be initiated if those KPIs indicate thresholds are likely to be exceeded.

Testing disaster recovery thoroughly helps you meet recovery objectives. Use automation to backup data. Regularly restore and recover to validate your recovery process and procedures.

These considerations improve the reliability of workloads and decrease business risk. Cloud change enablement practices should reflect this reduction in risk, and organizations should consider that because the risk is minimized, and reversible, they can be processed as standard changes.

Conclusion

Automation, workflow integration, and deployment tools in the cloud allow businesses to make small, frequent changes that optimize business risk and introduce business value at an increased rate. The change enablement processes should be adapted to reflect these objectives. For changes that do not take advantage of automation, consistency, or rollback, change approval can still be decentralized to the teams responsible for the workload.

If you do not update your change enablement practice, you may be introducing unwanted delays. Remember that the purpose of managing change is both to optimize business risk and increase business value.

Contributors

Contributors to this document include:

- Alex Livingstone, Principal WW CloudOps Specialist Solutions Architect, Amazon Web Services
- Rodney Lester, Principal Solutions Architect, Amazon Web Services
- Chris Kunselman, Senior Advisory Consultant, Amazon Web Services Proserve
- Peter Mullen, Senior Advisory Consultant, Amazon Web Services Proserve
- Michael Rhyndress, Senior DevSecOps Consultant, Amazon Web Services Proserve
- Diya Wynn, Global Readiness Lead, Amazon Web Services
- Swara Meghana Gattu, Cloud Architect, Amazon Web Services
- Yagya Vir Singh, Senior Technical Account Manager, Amazon Web Services

Further reading

For additional information, see:

- AWS Whitepapers & Guides
- AWS Well-Architected
- ITIL - What is IT Service Management?
- For more information on management and governance in the cloud, see ITIL®4 Acquiring and Managing Cloud Services and Management and Governance on AWS.

Document revisions

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

Change	Description	Date
Updated	Improved and updated based on the latest industry accepted practices of ITIL®4.	June 27, 2024
Updated	Updated for technical accuracy.	October 26, 2021
Initial publication	Whitepaper first published.	July 1, 2019

Note

To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser you are using.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

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