AWS Whitepaper

Change Management in the Cloud

Change Management in the Cloud: AWS Whitepaper

Table of Contents

Change Management in the Cloud

Publication date: **December 3, 2021** (*Document history*)

Abstract

Like every business function, change management should act as an enabler for your organization to succeed. Just as every business has some kind of finance function to ensure that it optimizes spending, change management is essential for optimizing business risk. You will benefit from change management if you have migrated to the cloud, have a hybrid environment, or are born in the cloud. An effective change management process enables agility and reduces time to market. It ensures that resources deliver business value, reduces failed changes, and helps ensure delivery to business. An effective record of change should also act as one of your first troubleshooting references when an incident occurs.

Research has found that high performance in terms of frequency of deployment, lead time for change, and change failure rate is correlated with version control, continuous delivery, and automated testing. (ITIL 4: High-Velocity IT, AXELOS, 2020, page 89)

This paper discusses change management, the Cloud Adoption Framework, and the Well-Architected Framework in the context of applying governance to deploying changes to your Amazon Web Services (AWS) environment.

Introduction

In a cloud computing environment, new IT resources are only a click away, which means you reduce the time to make those resources available to your developers from weeks to minutes. This results in a dramatic increase in agility for the organization, because the cost and time it takes to experiment and develop is significantly lower. Refer to 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. Stakeholders may have become accustomed to long release cycles using waterfall methodologies, and the transition to new ways of working that increase the frequency of releases can cause challenges. These challenges may result in increased stakeholder engagement, the introduction of unnecessary gates that hinder development progress, or unmanaged change.

Make frequent, small, reversible changes. Design workloads to allow components to be updated regularly. Make changes in small increments that can be reversed if they fail (without affecting customers when possible). For more information, refer to the AWS Well-Architected Framework.

Making frequent, small, reversible changes are essential to achieve agility, and are aligned to AWS best practices and strategies for designing and operating a cloud workload. Deployment of new services, software, patches, and configuration changes can all be automated, and they should still be governed by a change process.

In the cloud, you can enable this governance through policy and automation with a complete audit trail of the deployment steps. You can also preserve agility by “de-penalizing” the rollback of failed changes. In fact, to achieve agility, organizations must be willing to roll back changes that have adverse business consequences, and build the automation to make this happen.

Regular scheduled and unscheduled changes should flow through an unchanged pipeline that ensures all of your best practices are met before implementing a change in production. Different policies and procedures should exist for emergency changes, or changes that require manual processes during deployment.

What is ITIL?

The framework managed by AXELOS Limited defines an internationally recognized, best practice approach to IT Service Management (ITSM). Although it builds on ISO/IEC 20000 which "provides a formal and universal standard for organizations seeking to have their Service Management capabilities audited and certified," (*ITIL Service Operation, AXELOS, 2011, page 5*) ITIL goes one step further to propose operational processes to achieve the standard.

As defined by AXELOS, ITIL comprises five volumes that describe the ITSM lifecycle.

Table 1: ITIL volumes

ITIL volume	Description
Service strategy	The service strategy stage of the ITIL service lifecycle is crucial for defining an IT service strategy that operates effectively within its business context.
Service design	Through the service design stage of the service lifecycle, you can turn your strategy into a cost-effective plan that meets both current and future business needs.
Service transition	Best practice guidance on managing service change in a timely, cost-effective manner with minimal disruption to operations, customers, users, and the business.
Service operation	Best practice guidance on efficiently and effectively delivering these services for the benefit of the business, customers and users.
Continual service improvement	Best practice guidance on identifying and introducing a cycle of service management improvements, as well as a structured approach for assessing and measuring services.

AWS Cloud Adoption Framework

ITIL and AWS Cloud Adoption Framework (AWS CAF) are compatible. Like ITIL, AWS CAF organizes and describes all of the activities and processes involved in planning, creating, managing, and supporting modern IT services. It offers practical guidance and comprehensive guidelines for establishing, developing, and running cloud-based IT capabilities. The AWS CAF organizes guidance into six areas of focus, called perspectives.

Table 2: AWS CAF perspectives

Perspective	Description
Business	Helps you move from separate strategies for business and IT to a business model that integrates IT strategy. Agile IT strategies are aligned to support your business outcomes.
People	Helps Human Resources (HR) and personnel management prepare their teams for cloud adoption by updating staff skills and organizat ional processes to include cloud-based competencies.
Governance	Integrates IT Governance and Organizational Governance. It provides guidance on identifyi ng and implementing best practices for IT Governance, and on supporting business processes with technology.
Platform	Helps you design, implement, and optimize the architecture of AWS technology based on business goals and objectives. It helps provide strategic guidance for the design, principles, tools, and policies you use.
Security	Helps you structure the selection and implementation of controls. Following this guidance can make it easier to identify areas

Perspective	Description
	of non-compliance and plan ongoing security initiatives.
Operations	Helps you to run, use, operate, and recover IT workloads to levels that meet the requireme nts of your business stakeholders.

AWS CAF is an important supplement to enterprise ITSM frameworks used today because it provides enterprises with practical operational advice for implementing and operating ITSM in a cloud-based IT infrastructure. For more information, refer to the AWS Cloud Adoption Framework.

Change management in ITIL

In ITIL, change management is part of Service Transition. That is, the transitioning of an update or something new from Service Design to Service Operation. Change management aims to standardize the process for the efficient handling of all changes. In terms of a system or process, efficiency means maximizing productivity while minimizing wasted effort or cost.

A service change is defined by ITIL as “the addition, modification or removal of authorized, planned or supported service or service component and its associated documentation.” (ITIL Service Transition, AXELOS, 2011, page 43)

Change management is defined as “the Process responsible for controlling the Lifecycle of all Changes. The primary objective of Change Management is to enable beneficial Changes to be made, with minimum disruption to IT Services.” (ITIL Service Transition, AXELOS, 2011, page 229)

Change management is not designed to minimize business risk; the process should ensure that “overall business risk is optimized.” (ITIL Service Transition, AXELOS, 2011, page 43)

Assessing risk from the business perspective can produce a correct course of action very different from that which would have been chosen from an IT perspective, especially within high-risk industries. (ITIL Service Transition, AXELOS, 2011, page 54)

Every change should deliver business value; the change management processes should be geared towards enabling that delivery. ITIL states a number of benefits for effective change management, including “reducing failed changes and therefore service disruption, defects and re-work” (ITIL Service Transition, AXELOS, 2011, page 44) and “delivering change promptly to meet business timescales". (ITIL Service Transition, AXELOS, 2011, page 44)

In any environment, a good change management process should enable the delivery of business value while protecting the business by balancing risk against business value. It should do so in a way that maximizes productivity and minimizes wasted effort or cost for all participants in the process.

Change management in the cloud

It's key to remember that all changes should be delivering business value and that change management should be focused on optimizing business risk in a way that maximizes productivity while minimizing wasted effort or cost. The AWS Cloud enables automation that optimizes this business risk by:

- Minimizing the possibility of human error,
- Enabling the creation of identical environments for predictable and testable outcomes to changes,
- Removing the requirement to submit changes to scale infrastructure to meet business demand, and
- Automatically recovering from failure and rolling back failed changes.

The benefits of automation can dramatically reduce the business risk associated with change and increase business agility, ultimately delivering more business value (which is what change is about).

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

AWS Systems Manager Change Manager is an enterprise change management framework for requesting, approving, implementing, and reporting on operational changes to your application configuration and infrastructure. From a single delegated administrator account, if you use AWS Organizations, you can manage changes across multiple AWS accounts and across AWS Regions. Alternatively, using a local account, you can manage changes for a single AWS account. Use Change Manager for managing changes to both AWS resources and on-premises resources.

Configuration items in the cloud

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 would at least require an emergency change, and could put the business at risk for a significant amount of time. In the AWS Cloud, you can use Auto Scaling groups to automate this process. Failures can be automatically detected using predefined health checks, and servers can be automatically replaced with exactly the same configuration. This simple scenario shows the clear benefits of automation: human error is eliminated, configuration drift is eliminated, and business risk is minimized as the time to recover is dramatically reduced.

Auto Scaling groups can also be used to automatically provision additional resources to meet business demand. Again, in a traditional environment, the addition of servers may have required several business processes before approaching change management, and then a standard or normal change to implement the requirement. In the best case scenario, a lot of work was done to increase capacity; in the worst case scenario, the business was impacted and put at risk by all of the business processes required to introduce additional capacity, and it may not have been possible to meet the business demand in the timescales require.

With this example in mind, any manual approval steps required to recover from failure or scale capacity to meet business demand inherently introduce risk to the business. A change is considered to be the addition, modification, replacement, or removal of a configuration item. When approval is required to make a change to a configuration item, existing change management processes may forbid these automated scenarios. This scenario is where it may help to redefine what items are considered to be configuration items.

In the previous example, it is not the servers themselves that are the configuration items when they are in an Auto Scaling group, because they are transient and potentially non-configurable items. The Auto Scaling group and the image that is used to create the servers should be considered as the configuration items because they are the items that may put the business at risk if they are configured incorrectly.

To manage configuration items In the AWS Cloud, AWS Config can be used to assess, audit and evaluate the configuration of AWS resources allowing you to continuously monitor and record AWS resource configurations. With AWS Config, you can track the relationships among 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 would affect your other resources, which minimizes the impact of change-related incidents.

AWS CloudFormation change sets enable you to preview how proposed changes to a stack might affect your running resources; for example, to check whether your changes will delete or replace any critical resources. AWS CloudFormation makes the changes to your stack only after you decide to deploy the Change Set.

Automation

Another key consideration is understanding the business risk when deploying in the AWS Cloud. Regardless of whether or not a deployment is an application, a patch, or a configuration change, an optimized cloud configuration can automate the deployment process through an unchanged pipeline. This ensures repeatability and consistency across multiple environments, as well as enabling automation of software testing, compliance testing, security testing, and functional testing.

Although this does not guarantee against a change having an adverse impact, it does allow risks to be reduced, and those automated processes should not be reconsidered for every change. It is the actual configuration change itself that should have the focus.

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

AWS CodePipeline automates your software release process, enabling you to rapidly release new features to your users. With CodePipeline, you can quickly iterate on feedback and get new features to your users faster. Automating your build, test, and release process enables you to quickly and easily test each code change, and catch bugs while they are small and simple to fix. You can ensure the quality of your application or infrastructure code by running each change through your staging and release process.

EC2 Image Builder significantly 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 significantly reduces the risk of non-compliant images being used, as well as improving security, consistency and compliance.

Using AWS Systems Manager, you can automate operational tasks to help make your teams more efficient. With automated approval workflows and runbooks with rich text descriptions, you can 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, enabling you to incrementally roll out new changes and automatically halt the roll-out if errors occur.

Remediation

Changes should not be approved without considering the consequences of a failure. “Ideally, there will be a back-out plan, which will restore the organization to its initial situation” (ITIL Service Transition, AXELOS, 2011, page 48).

The AWS Cloud enables back-out plans to be fully automated using repeatable processes. Not all changes are reversible and “remediation may require a revisiting of the change itself in the event of failure” (ITIL Service Transition, AXELOS, 2011, page 48).

Deployments in the AWS Cloud that use an automated pipeline allow changes to be redeployed quickly and safely, minimizing risk and reducing business impact. In certain scenarios, it may not be possible to back-out changes or redeploy, in which case it might be that “it requires invoking the organization’s business continuity plan.” (ITIL Service Transition, AXELOS, 2011, page 48)

Even in the most severe cases, using continuous data protection in the cloud can enable sub-second recovery point objectives (RPOs), and recovery time objectives (RTOs) can be measured in minutes. Refer to CloudEndure Disaster Recovery for more information. Crucially, where it’s not possible to back out changes, the AWS Cloud provides methods to significantly reduce business risk and impact of a failed change by making it quicker and easier to redeploy or invoke 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 can then be switched to the new environment (green) while the old live environment (blue) remains available, but idle.

In this scenario, if a failure is discovered, users can be instantly redirected back to the blue environment, and the business impact is greatly reduced. It is also possible to combine this approach with a canary release that is easily enabled in the cloud. 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 for change management is the risk to the business of a change deployed in a manner like this is greatly reduced.

AWS CodeDeploy helps maximize your application availability during the software deployment process. It introduces changes incrementally and tracks application health according to configurable rules. Software deployments can easily be stopped and rolled back if there are errors.

AWS CloudFormation rollback triggers monitor the state of your application during stack creation and updating, and enable you to roll back that operation if the application breaches the threshold of any of the alarms you've specified. For each rollback trigger you create, you specify the CloudWatch alarm that AWS CloudFormation should monitor. AWS CloudFormation monitors the specified alarms during the stack create or update operation, and for the specified amount of time after all resources have been deployed. If any of the alarms goes to ALARM state during the stack operation or the monitoring period, AWS CloudFormation rolls back the entire stack operation.

AWS AppConfig supports best practices by rolling out configuration changes instantly or gradually. The configuration change is monitored over a time period that customers define. If you configure alarms in Amazon CloudWatch, AWS AppConfig can automatically rollback configuration changes in the event that those alarms are triggered.

Adapting change management to the cloud

There are two areas in which change process may need to be adapted. Because the risk and impact to the business of a failed change is greatly reduced, changes can be made more frequently and with more confidence in the rollback plan. As a result, the second area for consideration is the acceptance of rolling back changes.

If failed changes have a much lower impact due to the speed and consistency of roll back, activating roll backs should be considered to be part of the normal process. This is particularly true if it is possible to quickly remediate the issue and push it through the same automated pipelines to quickly deliver the original intended business value of the change.

With these considerations in mind, if automation, pipelines, and deployment methods are in place, it may be possible to reconsider the approach to standard changes. A standard change is where there is a defined trigger to initiate the change request. In addition, in a standard change, actions are well known, documented, and proven; authority is given in advance (or pre-authorized); and the risk is usually low. If the appropriate automation, testing, and deployment strategies are put in place, it should result in a scenario where large, infrequent, and risky changes are transformed in to small, frequent, low risk changes.

By understanding the risk-reduction strategies that are enabled by the AWS Cloud, it should be possible and it may even be necessary to widen the scope of a standard change to include deployments that would have previously been considered as normal changes due to the risks associated with them in traditional IT environments.

As changes become more frequent due to agile methodologies and increased automation, there is a risk that change management becomes overburdened with normal changes which can lead to delaying changes due to bandwidth constraints or, important details are missed as changes are not properly scrutinized due to resource constraints. Both of these scenarios introduce business risk which change management aims to optimize. In an environment of small, frequent changes, standard changes should become the new normal so proper scrutiny can be given to normal changes, optimizing business risk and enabling the delivery of business value.

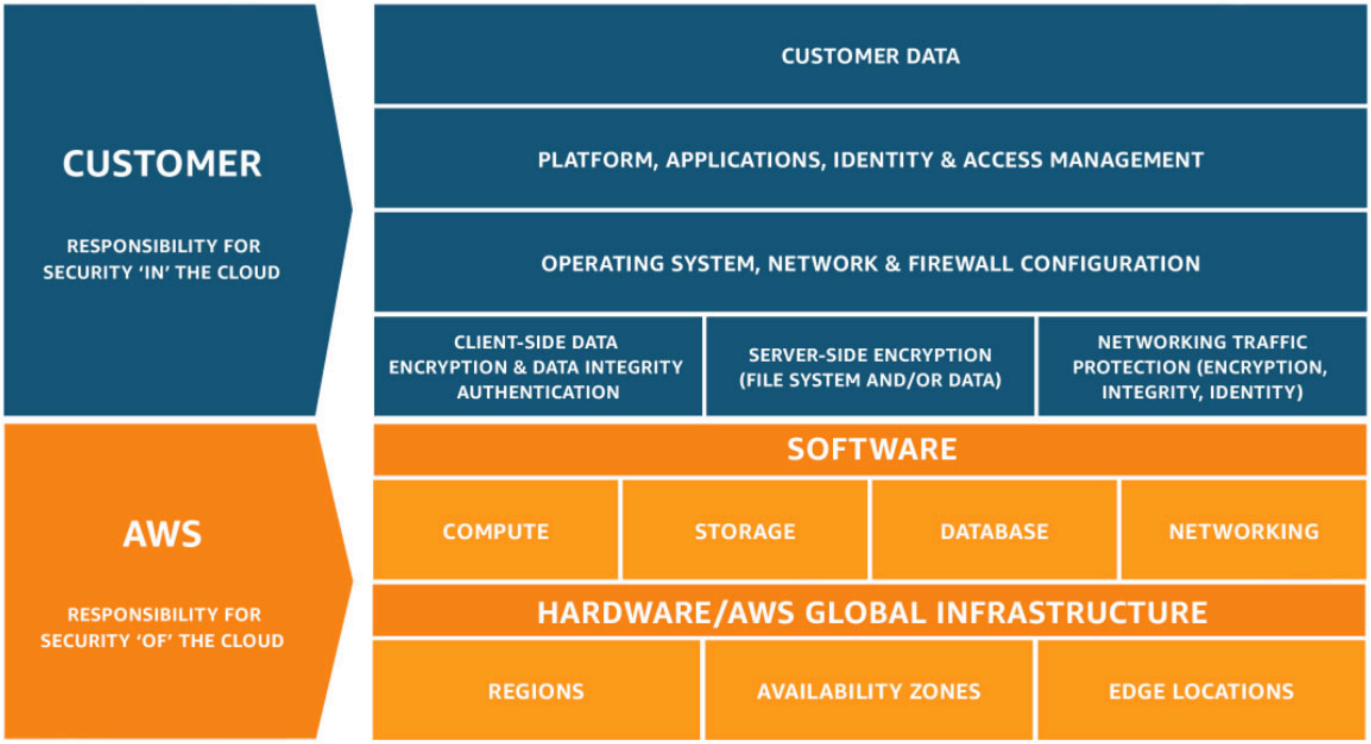
A reduction in the size of a change reduces the risk of disruption. Smaller changes also mean that change can happen more frequently. By changing more frequently, the organizations capability of changing is improved. Increased capability of change leads, in turn, to lower risk of disruption (ITIL 4: High-Velocity IT, AXELOS, 2020, page 89).

You can use deployment tools such as AWS CodePipeline, a continuous integration and continuous delivery service which define release process workflow. CodePipeline can pull source code directly from a repository, run builds and unit tests in AWS CodeBuild and then deploy changes into staging or production environment using AWS CodeDeploy if build and test cases are successful. CodePipeline automates the entire release process which reduces manual errors, increasing agility and business value allowing you to optimize business risk.

Service transition

After a release has been approved via the change management process and all the appropriate project management, release, and deployment management steps have been followed, the release is deployed and enters into a process of service validation and testing.

It's worth pausing here to 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.


CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

Shared Responsibility Model

As previously stated, 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 will lead to an increased requirement 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 releases meet customer expectations and that IT operations teams are able to support this new added business value. The criteria for assessing this value in the cloud should not change from what already exists, but the organization must be prepared for the increase in releases and adapt the implementation of these processes by introducing automation to the processes.

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 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, this must be tracked with monitoring and metrics on the primary business function of the services. Separate service level requirements may exist for different aspects of services, and additional dimensions may be required as metrics to indicate which aspect is being measured. Indeed, it is often this monitoring that drives an automated rollback, if it indicates that there is a trend towards violating an SLA.

Operations must be able to support a new release or service before it is made available to the customer. With the correct tooling, this process can be largely automated by automating the creation of documentation, provisioning automated runbooks and playbooks, and building predefined and automated patching plans. This process can be made even more robust by using the correct tooling to ensure that only pre-approved services are used.

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

In the AWS Cloud, 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 get a unified view of operational health. 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.

CloudWatch provides different features, including Dashboards, Synthetic monitoring, CloudWatch Application Insights, and ServiceLens, which can be used during and after Service Transition to ensure that actionable alarms are present to prevent or remediate against service degradation or failure.

Access to metrics, logs and dashboards to monitor the health of application helps teams to resolve problem faster, reducing business risk.

Reliability

Change implementation has a direct impact on the availability of workloads, and the ability to recover from logical disasters. There is detailed information in the *AWS Well-Architected Framework Reliability Pillar* whitepaper, specifically in the 'Operational Considerations for Availability' section. Automation of change is foremost in maximizing availability. If you have any manual processes, you lose critical time awaiting those manual actions.

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

Test disaster recovery thoroughly to ensure that recovery objectives are met. All backing up of data must be done through automation. Regularly restore and recover in order to ensure that your recovery process and procedures are valid.

These considerations improve the reliability of workloads and decrease risk. Change management processes should reflect this reduction in risk and organizations should consider that because the "risk is usually low and always well understood," (ITIL Service Transition, AXELOS, 2011, page 48) automated, frequent, small, and reversible changes can be processed as standard changes.

Conclusion

Automation, integration, and deployment tools in the cloud allow the business to make small, frequent changes that reduce business risk and introduce business value at an increased rate. Change processes should be adapted to reflect what is actually being changed; the increase in the amount of change and the reduced risk associated with these changes. For changes that do not take advantage of automation, consistency, or rollback, the change process should remain as is.

Finally, it’s always worth considering the business impact and risk of not implementing a change or introducing delay, and remembering that the purpose of managing change is to optimize business risk.

Contributors

Contributors to this document include:

- Alex Livingstone, Cloud Operations Specialist, Amazon Web Services
- Rodney Lester, Reliability Lead, Well-Architected, Amazon Web Services
- 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 history

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

Change	Description	Date
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.