

What is Configuration Management? A Comprehensive Guide



In this e-guide

- What is configuration management? A comprehensive guide p. 2
- How does configuration management work? p. 3
- History of configuration management p. 5
- Key benefits of configuration management p. 7
- Risks of configuration management p. 10
- Change and asset management p. 11
- Software configuration management p. 13
- Configuration management in DevOps p. 15
- Configuration management tools p. 19
- How to get started with configuration management implementation p. 21
- The future of configuration management p. 23
- Further reading p. 25










In this e-guide:

Organizations rely on configuration management (CM) because it enables administrators and software developers to understand how a change to one configuration item (CI) will affect other items.

For business leaders, CM is a valuable instrument in business governance and compliance efforts.

We've created this expert guide to arm you with everything you need to know about configuration management. From software and tools, risks to look out for and what the future holds for CM – this 25-page guide has it all.

In this e-guide

- 
[What is configuration management? A comprehensive guide](#)
p. 2
- 
[How does configuration management work?](#)
p. 3
- 
[History of configuration management](#)
p. 5
- 
[Key benefits of configuration management](#)
p. 7
- 
[Risks of configuration management](#)
p. 10
- 
[Change and asset management](#)
p. 11
- 
[Software configuration management](#)
p. 13
- 
[Configuration management in DevOps](#)
p. 15
- 
[Configuration management tools](#)
p. 19
- 
[How to get started with configuration management implementation](#)
p. 21
- 
[The future of configuration management](#)
p. 23
- 
[Further reading](#)
p. 25

What is configuration management? A comprehensive guide

Margaret Rouse, WhatIs.com

Configuration management (CM) is a [governance](#) and [systems engineering](#) process used to track and control IT resources and services across an enterprise. When properly implemented, configuration management ensures that an organization knows how its technology assets are configured and how those items relate to one another.

The CM process seeks to identify and track individual configuration items (CI), and document their functional capabilities and interdependencies. A CM tool helps a business enforce a desired configuration state for each item and provides timely alerts of any configuration problems.

Organizations rely on configuration management because it enables administrators and software developers to understand how a change to one CI will affect other items. For business leaders, configuration management is a valuable instrument in business governance and compliance efforts.

Configuration management is typically implemented in the form of software tools, but it is a broad approach to systems engineering and governance, and it can be codified in standardized frameworks. For example, the IT Infrastructure Library ([ITIL](#)) v3

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

framework includes a detailed treatment of service asset and configuration management.

The roles and uses of configuration management have evolved and expanded over time. Today, the process has moved beyond the traditional management of physical enterprise compute, storage and network hardware to embrace ever-advancing practices such as software-driven infrastructures, software configuration management and even DevOps practices.

How does configuration management work?

For a configuration management system to operate, it needs some form of mechanism in which to store the information it governs. Originally, this was called the [configuration management database \(CMDB\)](#); ITIL v3 introduced the concept of a configuration management system (CMS) to replace the CMDB. The CMDB promotes the concept of a singular monolithic repository, while the CMS provides a conceptualized system of CMDBs that act together to support the needs of this governance process. Both demonstrate advantages over a static CM spreadsheet or a text file that requires significant manual upkeep and cannot integrate base workflows and best practices.

Every service management tool is deployed with a supporting data repository. Without the governance process of configuration management validating its contents, the repository is simply an operational database with unverified data, not a CMDB or CMS.

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25










Automated configuration audit and verification components entitle a repository to be leveraged as an authorized gold source of assets. A manual audit is also possible.

A CM process and its supporting repository, CMDB or CMS, face the challenge of overlapping and contradicting data from sources across the enterprise. A configuration management plan must include a way to merge and reconcile CIs to present a single point of reference or sole source of truth.

As the CMDB grows and contains more configuration items, it becomes possible to predict the impact of configuration changes, a key role in change management. By tracking dependencies, for example, administrators can determine the impact that a hardware, software, network or other outage might have on other systems or resources.

Even when configurations are well documented and carefully enforced, configuration management must account for the reality of periodic changes, such as software upgrades and hardware refreshes. Infrastructure and architectural changes may be required to tighten security and enhance performance. This makes [change requests](#) integral to the CM practice. This might be as simple as opening a certain port on a firewall to accommodate an application's new feature, or relocating one or more busy servers on the local network to improve performance of other applications on the subnet.

In this e-guide

- 
[What is configuration management? A comprehensive guide](#)
p. 2
- 
[How does configuration management work?](#)
p. 3
- 
[History of configuration management](#)
p. 5
- 
[Key benefits of configuration management](#)
p. 7
- 
[Risks of configuration management](#)
p. 10
- 
[Change and asset management](#)
p. 11
- 
[Software configuration management](#)
p. 13
- 
[Configuration management in DevOps](#)
p. 15
- 
[Configuration management tools](#)
p. 19
- 
[How to get started with configuration management implementation](#)
p. 21
- 
[The future of configuration management](#)
p. 23
- 
[Further reading](#)
p. 25

History of configuration management

In the broadest sense, configuration management traces its roots to the 1960s, when the U.S. Defense Department instituted military standards for general hardware and configuration management. Standards evolved and consolidated into what became ANSI/EIA-649-1998.

The basic CM model was adapted and implemented for myriad technical disciplines, including systems engineering, product lifecycle management and application lifecycle management, as well as later standards such as ISO 9000, COBIT and Capability Maturity Model Integration (CMMI). The ITIL framework, which emerged in the 1980s, introduced principles and practices for enterprises to select, plan, deliver and maintain IT services. These allowed IT to function as a business service rather than simply a cost center -- a concept that continues to resonate today. ITIL has embraced configuration management as a central part of its framework through its most recent update to [ITIL v4](#) in 2019 and 2020.

IT and business leaders readily adopted configuration management with the explosion of enterprise computing in the 1970s and 1980s. Data center operators realized that standardized practices were vital to the established functionality of servers and systems within a production environment. IT further refined the CM process to include specific activities such as change control or change management to ensure that changes were documented and validated.

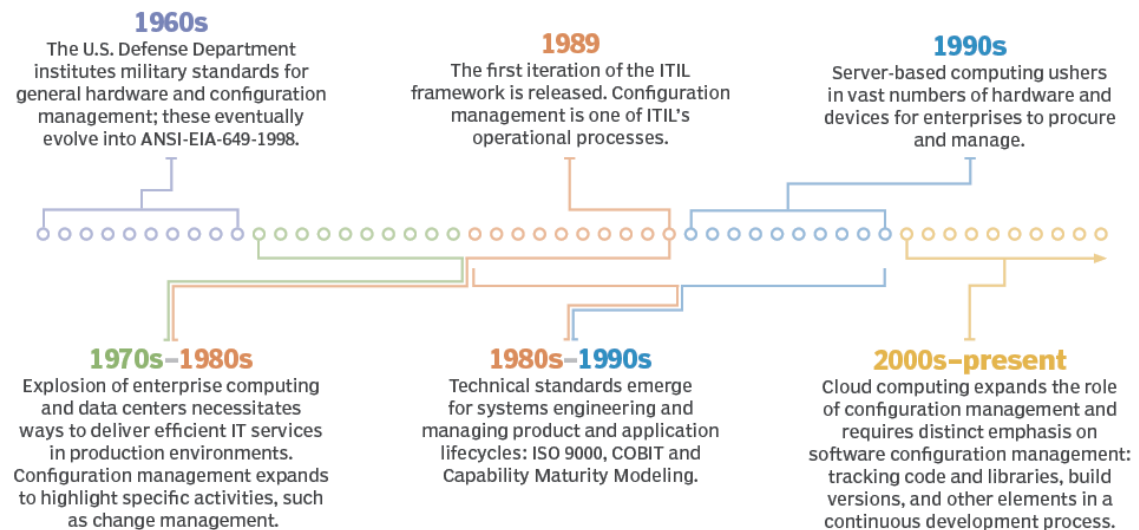
In this e-guide

- What is configuration management? A comprehensive guide p. 2
- How does configuration management work? p. 3
- History of configuration management p. 5
- Key benefits of configuration management p. 7
- Risks of configuration management p. 10
- Change and asset management p. 11
- Software configuration management p. 13
- Configuration management in DevOps p. 15
- Configuration management tools p. 19
- How to get started with configuration management implementation p. 21
- The future of configuration management p. 23
- Further reading p. 25









The broad shift from mainframes to server-based computing in the early 1990s multiplied the volume of hardware and devices in the data center. A centralized mainframe gave way to racks of individual servers, storage subsystems, networking gear and appliances, as well as full-featured endpoint systems such as desktop PCs.

Configuration management

The uses of configuration management have evolved and expanded beyond enterprise computing systems and data centers into modern IT areas of cloud and software-defined environments.



In this e-guide

- 
[What is configuration management? A comprehensive guide](#)
p. 2
- 
[How does configuration management work?](#)
p. 3
- 
[History of configuration management](#)
p. 5
- 
[Key benefits of configuration management](#)
p. 7
- 
[Risks of configuration management](#)
p. 10
- 
[Change and asset management](#)
p. 11
- 
[Software configuration management](#)
p. 13
- 
[Configuration management in DevOps](#)
p. 15
- 
[Configuration management tools](#)
p. 19
- 
[How to get started with configuration management implementation](#)
p. 21
- 
[The future of configuration management](#)
p. 23
- 
[Further reading](#)
p. 25

Configuration management practices continue to evolve and now embrace remote resources and services. For example, cloud users employ these practices and tools to monitor the resources, services and workloads deployed within the public cloud.

In addition, configuration management has bifurcated into software development efforts, helping developers track software components, libraries, build versions and other software elements used in Agile or continuous development initiatives. This dovetails with DevOps and related areas such as [infrastructure as code](#) (IaC).








Key benefits of configuration management

From its earliest implementations in IT, configuration management served four key purposes: consistency, security, service delivery and compliance.

Although CM tools have become more sophisticated and the practices have expanded in scope, those four central purposes remain as important as ever.

Consider a traditional data center filled with dozens or hundreds of physical servers, network switches, storage devices and so on. It is vital that IT staff and business leaders understand precisely what is present in the environment and ensure that every device, OS and application is configured in a known and acceptable way. They might even implement measures to enforce an established configuration. Consequently, configuration management **provides an underlying consistency** to the IT environment.

In this e-guide

-  What is configuration management? A comprehensive guide p. 2
-  How does configuration management work? p. 3
-  History of configuration management p. 5
-  Key benefits of configuration management p. 7
-  Risks of configuration management p. 10
-  Change and asset management p. 11
-  Software configuration management p. 13
-  Configuration management in DevOps p. 15
-  Configuration management tools p. 19
-  How to get started with configuration management implementation p. 21
-  The future of configuration management p. 23
-  Further reading p. 25

When a device requires service or replacement, an established configuration provides a baseline that can be preserved and applied to replacement devices.

Data centers have long relied on hardware and software configurations to help **support enterprise security**. Configurations affect security in countless ways -- login credentials are required, periodic password changes are mandated, firewall ports open and close, network subnets are established, etc. Documentation and consistency through configuration management help establish and maintain security.

Consistency -- doing the same things in the same ways -- also plays directly into quality-of-service (QoS) and [service delivery](#), and configuration management holds a core role here. **Service delivery** helps to ensure that the environment (and the hardware and software operating therein) operates in a known and validated manner. IT service management (ITSM) frameworks such as ITIL take the concept even further by outlining the processes, people and products involved in service delivery. IT administrators use configuration management and ITSM to enforce accepted approaches while guarding against prohibited approaches. The goal is to ensure that any services provided by IT are available, reliable and secure.

Finally, configuration management is a major factor in [regulatory compliance](#). Compliance dictates adherence to guidelines, specifications or actions established by a governing authority, be it a recognized standards body (such as ANSI or ISO), an industry organization or a government. Configuration management by itself does not demonstrate compliance, but the ability of an organization to show that a CM

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

mechanism is in place to discover, preserve, enforce and audit a configuration across the infrastructure can [support the organization's compliance efforts](#).

What are the risks of not using configuration management?

Foregoing the benefits of consistency, security, service delivery and compliance support that configuration management can provide, an enterprise that operates without a CM plan invites many areas of risk.

Imagine an IT environment without configuration management: A data center operates workloads, but IT administrators and business leaders have no single source of truth about the hardware and software makeup across the enterprise. Staff know some of the hardware and software setups but must directly inspect each data center element to determine the existing configuration -- and there is no common method to determine what a setup *should* be.

- With no clear and complete index of hardware and software, IT can't be certain what is present and running in the environment.
- Everything must be done manually -- identifying hardware and software elements and reviewing and managing their configuration -- which is a practical impossibility for modern IT environments.
- With no established documented configuration standard, an organization cannot ensure proper security, service performance or compliance without undertaking time-consuming manual audits.

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

Risks of configuration management








Although the benefits of configuration management can be compelling, the technology is not perfect. CM platforms and practices pose challenges, starting with adoption and integration.

The CM process requires an organization to identify each element, understand its specific configuration details, enter those details accurately into a documentation platform and then manage that data. Deciding which configuration data to collect and how to manage that data over time -- especially as hardware and software changes are required -- places a demand on IT staff.

Older CM platforms were little more than spreadsheets that required extensive manual effort to populate and manage. CM platforms later provided ever-greater levels of discovery and automation to help organizations fill in the blanks of each element's configuration and dependencies. But equipment that is not discoverable by or suited for a specific CM tool may require additional tools, spreadsheets or other documentation. This effectively splits the CMDB and carries a significant risk of configuration management errors and oversights.

Effective configuration management must be an all-or-nothing effort. Overlooked hardware and software narrow the view of CM processes, reducing the IT staff's ability to manage the environment. One forgotten desktop with an unpatched OS can expose the entire business to catastrophic security vulnerabilities.

In this e-guide

- 
[What is configuration management? A comprehensive guide](#)
p. 2
- 
[How does configuration management work?](#)
p. 3
- 
[History of configuration management](#)
p. 5
- 
[Key benefits of configuration management](#)
p. 7
- 
[Risks of configuration management](#)
p. 10
- 
[Change and asset management](#)
p. 11
- 
[Software configuration management](#)
p. 13
- 
[Configuration management in DevOps](#)
p. 15
- 
[Configuration management tools](#)
p. 19
- 
[How to get started with configuration management implementation](#)
p. 21
- 
[The future of configuration management](#)
p. 23
- 
[Further reading](#)
p. 25

Data sharing, integrity and protection are also vital. The configuration of an enterprise IT infrastructure involves sensitive details, such as a server's IP address. That information must be kept secure, yet it also must be available to other stakeholders such as corporate compliance officers who perform audits. Deciding which stakeholders or staff can access and modify CM data is a delicate matter.

The evolution of data center technologies poses challenges for configuration management. Consider [IaC pipelines](#) where fully virtualized data center resources are pooled, provisioned and managed as virtual machines, containers, cloud instances and other virtual constructs. Configuration management is essential to manage virtualized environments, but mistakes in defining virtual configurations can lead to new security risks and wasted resources (sprawl).

Change and asset management

The discussion of configuration management typically involves the ideas of change management and asset management. These ideas complement configuration management, but it's important to understand the differences.

[Change management](#) is a process that directs the organization's approach to change. This applies to business goals and workflows, helping employees and customers to accommodate and adapt to change. But IT also employs change management to formalize the approach to change within the data center or enterprise computing environment. This may include protocols for how changes are requested and how to

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

define procurement, deployment and setup/configuration. A company will also want to organize change approvals and validations.








For instance, consider a workload running on a server. Application updates aren't applied to the environment indiscriminately -- an unmanaged change will result in a configuration change that might affect the production environment. Change management is the process by which the new version might be identified, considered, tested, approved for deployment, deployed, configured and validated. Changes must be registered in the CMS. Change management might also detail any training or guidelines needed for users to help smooth a transition or minimize disruption.

Consequently, [change management is different than configuration management](#), but managing change -- and the potential impacts of change on a production environment -- is a vital part of configuration management. The detail and formality involved in a change management process can vary depending on the size and type of organization; a large and highly regulated business will typically use a detailed change management process.

IT [asset management](#) shares the common use of data to identify the presence of hardware, software and other tangible assets across the business. Asset management is generally focused on business and accounting applications.

For example, asset management is aware of the servers, storage, networking gear, endpoint devices and other IT assets across the enterprise. Still, it is more concerned with the cost and validity of licenses, the physical location and costs of each asset, and

In this e-guide

- 
[What is configuration management? A comprehensive guide](#)
p. 2
- 
[How does configuration management work?](#)
p. 3
- 
[History of configuration management](#)
p. 5
- 
[Key benefits of configuration management](#)
p. 7
- 
[Risks of configuration management](#)
p. 10
- 
[Change and asset management](#)
p. 11
- 
[Software configuration management](#)
p. 13
- 
[Configuration management in DevOps](#)
p. 15
- 
[Configuration management tools](#)
p. 19
- 
[How to get started with configuration management implementation](#)
p. 21
- 
[The future of configuration management](#)
p. 23
- 
[Further reading](#)
p. 25

how those things are procured and ultimately disposed of. By comparison, configuration management is typically concerned only with an asset while it is in operation.

As another example, consider a server that is procured for a specific purpose. That server will be subject to both [asset management](#) and [configuration management](#). If that server is repurposed at some point in its lifecycle, it will take on a new entry in configuration management because the server would be configured differently to do a different job. But, it would remain the same business asset under asset management.

Software configuration management

Configuration management has expanded into the realm of software development and deployment, where it is known as software configuration management (SCM) or unified configuration management. While it is rooted in broader configuration management, SCM focuses on the management of the many artifacts and components involved in software development, including source code, modules, libraries and APIs, along with related elements such as documentation, change requests and trouble tickets.

SCM can be used across the entire [software stack](#). It provides additional structure and standardization to the software development process, which can help organizations establish baselines, enhance and organize reporting, manage changes and oversee

In this e-guide

- What is configuration management? A comprehensive guide p. 2
- How does configuration management work? p. 3
- History of configuration management p. 5
- Key benefits of configuration management p. 7
- Risks of configuration management p. 10
- Change and asset management p. 11
- Software configuration management p. 13
- Configuration management in DevOps p. 15
- Configuration management tools p. 19
- How to get started with configuration management implementation p. 21
- The future of configuration management p. 23
- Further reading p. 25

resources allocated to a development project. SCM is codified in standards such as IEEE 828, which was updated in 2012.

Benefits of software configuration management

Software configuration management (SCM) provides several benefits to organizations seeking more control over their software development process, from source codes to APIs to change requests.

Manage processes

SCM enables a team to create and enforce processes. This can be particularly important for highly regulated businesses or sophisticated projects. Process management can extend to guidelines for developer collaboration.

Manage builds

SCM can guide the software development toolchain used to build and test software elements. For example, build management might dictate the specific tools to be used as well as how each tool should be set up and used.

Establish baselines

Baselines may involve the specification of minimum component versions, required tool versions and setups, testing protocols and



procedures. Baselines can also extend to the host environment, stipulating the hardware and software environment needed. For example, if the software will operate on a specific processor and OS, such elements would be part of the baseline environment for testing.

Track tickets

With its formalized ticketing and reporting processes, SCM enables an

organization to fully trace and document every defect or incident.









Control changes

SCM can be a fundamental part of change control in software development, helping development teams to receive, approve, plan, implement and validate change requests received from stakeholders across the enterprise.

Audit and account for changes

SCM shows developers and stakeholders all of the information related to the development process. Auditing helps to ensure that a software configuration includes all of its intended parts, and it can provide configuration summaries that reconcile specifications, components, documents and requirements.

In this e-guide

- 
[What is configuration management? A comprehensive guide](#)
p. 2
- 
[How does configuration management work?](#)
p. 3
- 
[History of configuration management](#)
p. 5
- 
[Key benefits of configuration management](#)
p. 7
- 
[Risks of configuration management](#)
p. 10
- 
[Change and asset management](#)
p. 11
- 
[Software configuration management](#)
p. 13
- 
[Configuration management in DevOps](#)
p. 15
- 
[Configuration management tools](#)
p. 19
- 
[How to get started with configuration management implementation](#)
p. 21
- 
[The future of configuration management](#)
p. 23
- 
[Further reading](#)
p. 25

Organizations frequently undertake multiple projects, each involving myriad components and multiple developers or teams. Without some cohesive way to bring order to the process, software building and testing would devolve into chaos. When several developers work on the same source code at the same time, the various changes won't integrate well and the software will essentially break. To avoid that outcome, SCM creates multiple lines of development and reconciles each line into a final source for a build, allowing many developers to work on the same code simultaneously.








Configuration management in DevOps

Configuration management is crucial to collaborative and rapid software development paradigms such as [DevOps](#). With CM, software developers can create, test and deploy builds with minimal IT oversight.

Components must be brought together and integrated into a build, and each resulting build also carries unique version designations before being tested and deployed. Configuration management tracks the components and ensures that a desired build uses only certain components.

Moreover, each build must be thoroughly tested, and configuration management can be employed to specify the tools and test files needed to validate a given build. When paired with automation, CM techniques can accelerate testing and release processes.

In this e-guide

- 
[What is configuration management? A comprehensive guide](#)
p. 2
- 
[How does configuration management work?](#)
p. 3
- 
[History of configuration management](#)
p. 5
- 
[Key benefits of configuration management](#)
p. 7
- 
[Risks of configuration management](#)
p. 10
- 
[Change and asset management](#)
p. 11
- 
[Software configuration management](#)
p. 13
- 
[Configuration management in DevOps](#)
p. 15
- 
[Configuration management tools](#)
p. 19
- 
[How to get started with configuration management implementation](#)
p. 21
- 
[The future of configuration management](#)
p. 23
- 
[Further reading](#)
p. 25

On the operations side, configuration management enables developers to stipulate an appropriate deployment environment for a build. This might involve creating and configuring VMs, but it's also a powerful way to create and configure containers using container management/orchestration tools such as Kubernetes.

Every step of the way, configuration management provides logging to help track changes, audit access, enforce established configurations (such as specific version components) and maintain security within the library of components and builds.

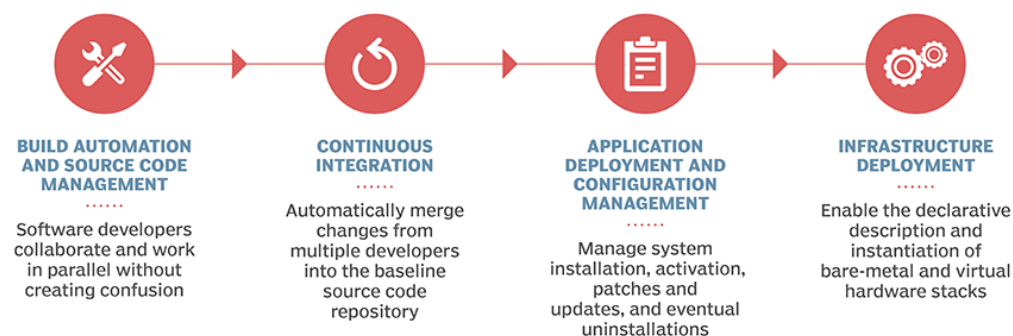
CI/CD and configuration management: CM is used throughout the CI/CD toolchain. With build automation and source code management, CM can establish and enforce version control and change tracking. It also assists with version control and change tracking to log any activity within the repository and enforce any version restrictions.

Application deployment and configuration management define and enforce the resources needed to run the build in a desired configuration. The final step is actual deployment, where the build is delivered for signoff to deployment -- or automatically deployed to live servers -- and connected to running services as desired. CM tools such as [Ansible](#), [Puppet](#), [Chef](#) and [SaltStack](#) are generally geared toward the latter part of the CI/CD toolchain, where workloads are deployed into the data center's hardware environment.

In this e-guide

- What is configuration management? A comprehensive guide p. 2
- How does configuration management work? p. 3
- History of configuration management p. 5
- Key benefits of configuration management p. 7
- Risks of configuration management p. 10
- Change and asset management p. 11
- Software configuration management p. 13
- Configuration management in DevOps p. 15
- Configuration management tools p. 19
- How to get started with configuration management implementation p. 21
- The future of configuration management p. 23
- Further reading p. 25

Links in the DevOps toolchain



Configuration management and infrastructure as code: IaC principles rely on extensive virtualization to discover and pool resources across the data center environment, then provision and manage those resources based on software-driven or software-defined actions. DevOps is a principal driver of IaC because the use of code to create and manage infrastructure for a build's deployment is a natural extension of the software development and testing processes. IaC carries code into the operations side and allows developers to readily deploy builds without ever touching or configuring actual hardware.

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

IaC description files can be written, tested, validated, version controlled and deployed much like any other software. This also means various [testing processes are essential for an IaC deployment](#). These include static tests, unit tests, system tests, integration tests and blue/green (or A/B) tests. [IaC security](#) also must be considered.

Configuration management relies on programmatic techniques (instructions or code) to create, test and deploy software builds and manage infrastructure. Writing the code to drive CM processes in DevOps can use either *imperative* or *declarative* programming techniques.

Imperative programming focuses on *how* something is done. It tends to be more literal and detailed -- for example, if someone asks you for directions to a Best Buy store, the imperative response would include all of the specific turns and distances to reach the destination. By comparison, *declarative* programming focuses on *what* is being done. It is more focused on ultimate goals. To get to a Best Buy store, the declarative response would be to provide the store's address -- it doesn't matter much which route a person takes to get there. Configuration management embraces declarative programming, which allows goals to be stated more clearly and concisely.

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

Configuration management tools

A wide array of tools are available to address CM tasks, which include:

- **Discovery.** The tool detects hardware and software present within the management scope, such as the data center. It collects relevant information about the CI and enters that information into a database.
- **Configuration states.** The tool establishes and enforces desired configuration states for selected hardware or software CIs. This is usually accomplished using policies and automation. Deviations from a desired state (baseline) are alerted and logged, allowing administrators to investigate and remediate unauthorized change attempts.
- **Version control.** Tools ensure that specific software versions or components are deployed or built. In software development, version control ensures that a build is assembled from specific components.
- **Change control.** The tool controls the configuration and implements a process that coordinates, authorizes, documents and reports authorized changes within the management scope.
- **Auditing.** The tool scans the environment, validates that current configurations are in place, identifies and reports any dependencies and ensures that the overall environment is configured as expected.

Because CM tools are so different, [selecting the right configuration management tool](#) can make or break a CM initiative. Among the numerous features and attributes to consider are:

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

- **Flexibility.** The tool should be easy to use, extensible and easily integrated with other tools, such as those used for systems management or the help desk. A tool that's flexible can discover and manage more than one that isn't, and this should result in an organization needing to deploy a small number of CM tools.
- **Comprehensive reporting.** Activity logging and reporting enables administrators and auditors to gather a complete picture of the environment and any changes made to it.
- **Collaboration and communication.** The tool should bring configuration management together with other management functions so that administrators are notified of change requests, change attempts and completed changes -- especially changes impacting security or compliance.
- **Scalability and extensibility.** Simple tools should be adequate for small to mid-sized organizations. Larger organizations, however, will need sophisticated tools that support complex and expanding environments.
- **Cloud support.** The management scope of the CM process is growing. Environments can be heterogeneous or homogeneous. Single data centers are now often complemented by secondary data centers, private cloud deployments and public cloud infrastructures. Tools must be able to operate across multiple environments.
- **Cost.** Tools can be expensive, but the cost is readily justified by the risks that a tool mitigates.

Leading configuration management tools vary in their scope and purpose. General tools such as SolarWinds Server Configuration Monitor, CFEngine, [Puppet](#), Chef, Ansible, SaltStack, Juju and Rudder can handle data center hardware and SCM with some degree of automation. Note that there is active industry consolidation around CM vendors, which may complicate product availability and roadmaps: In October

In this e-guide

What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

2020, VMware acquired SaltStack and Progress Software acquired Chef, while Ansible is now part of IBM Red Hat.

Software developers can use many of these tools for SCM tasks, but will also employ other tools such as Atlassian Bamboo for continuous delivery and release management; JetBrains TeamCity, Jenkins, and Apache Maven for configuration management and continuous integration; Git, Apache Subversion, and Mercurial for source code management; simple Wikis for documentation; and JFrog Artifactory, Sonatype Nexus and Apache Archiva for artifact repositories used in version control.

How to get started with configuration management implementation

Configuration management is a core part of any IT organization, but it takes careful planning and ongoing effort to successfully deploy it across an enterprise. The distinct [steps to a CM deployment](#) are:

1. **Establish a baseline.** In the discovery phase of configuration management, a tool or platform examines the IT environment, queries and discovers the hardware, software, services and other elements of the environment. This process establishes a current baseline of the elements, along with the configurations and dependencies of the elements. An organization will then know what is present and how everything works. The baseline is the foundation for change and change management. Consequently, a tool must see the entire environment, including physical, virtual and sometimes even cloud elements.

In this e-guide








What is configuration management? A comprehensive guide	p. 2
How does configuration management work?	p. 3
History of configuration management	p. 5
Key benefits of configuration management	p. 7
Risks of configuration management	p. 10
Change and asset management	p. 11
Software configuration management	p. 13
Configuration management in DevOps	p. 15
Configuration management tools	p. 19
How to get started with configuration management implementation	p. 21
The future of configuration management	p. 23
Further reading	p. 25

Tools rarely identify every detail about every element, so IT staff must fill in the gaps or uncertainties in the baseline; this CM phase may take considerable time.

2. **Maintain that baseline.** Once a baseline is established, it must be actively maintained. The tool will typically detect, log and report changes -- all of which must be approved and documented. Change and change management should follow clearly defined policies and practices to prevent accidental, unapproved or ad hoc changes. Tools with strong CM enforcement features can actively prevent changes until approval and validation steps are completed. Changes that do take place must be carefully documented to keep the CM tool current. Many CM initiatives fail when the environment deviates from the baseline.
3. **Audit the database.** Even with active maintenance and comprehensive policies to govern CM processes, the tool and the environment must still be subjected to periodic auditing to validate the configuration and ensure that the current environment matches the current baseline. Any differences must be corrected, and the reasons behind any differences must be understood and remediated. For example, if a certain server remains undetected, it may need an updated CM agent or other fix.
4. **Test to ensure your tool performs as intended.** Finally, it's important to use the CM tool productively. For example, direct the CM tool to implement an OS patch or driver upgrade across a subset of the environment. Validate that the tool can accomplish such tasks within change management guidelines. Provide a means for rolling back undesired changes.

Configuration management heavily depends on policy, process and automation, which must be integrated into the CM tool or platform. But these three factors are not single steps. Just as a configuration must be regularly revisited, audited and tested, so must

In this e-guide

- 
[What is configuration management? A comprehensive guide](#)
p. 2
- 
[How does configuration management work?](#)
p. 3
- 
[History of configuration management](#)
p. 5
- 
[Key benefits of configuration management](#)
p. 7
- 
[Risks of configuration management](#)
p. 10
- 
[Change and asset management](#)
p. 11
- 
[Software configuration management](#)
p. 13
- 
[Configuration management in DevOps](#)
p. 15
- 
[Configuration management tools](#)
p. 19
- 
[How to get started with configuration management implementation](#)
p. 21
- 
[The future of configuration management](#)
p. 23
- 
[Further reading](#)
p. 25

the surrounding CM policy, process and automation elements be reviewed and updated on a regular basis to ensure that the tool (and its usage) remains consistent with IT and business goals.






The future of configuration management

One of the greatest drivers for tomorrow's CM model lies in software-defined environments. More of the enterprise IT environment uses virtualization, automation and management to provision, deploy and manage resources and services through software. With the rise of data center technologies such as software-defined storage, software-defined networking, SDDC and IaC, future CM tools and practices must be able to discover and interoperate with flexible and virtual software environments.

Consider the impact of IaC. Resources are typically virtualized and pooled, and a predefined set of instructions is used to provision those resources, configure the instance, deploy a workload into the instance, connect associated services such as load balancers or storage, configure the workload and manage the deployment over time. Configuration management must be able to discover new deployments as they appear, and integrate those new instances into reporting and change management oversight. This means CM tools must be able to add, remove and manage instances ad hoc.

Another emerging technology to consider is [GitOps](#), which enables a data center team to deploy container clusters using the Git code management and version control system. This effectively merges the use of containers, software development

In this e-guide









-  What is configuration management? A comprehensive guide p. 2
-  How does configuration management work? p. 3
-  History of configuration management p. 5
-  Key benefits of configuration management p. 7
-  Risks of configuration management p. 10
-  Change and asset management p. 11
-  Software configuration management p. 13
-  Configuration management in DevOps p. 15
-  Configuration management tools p. 19
-  How to get started with configuration management implementation p. 21
-  The future of configuration management p. 23
-  Further reading p. 25

paradigms and SDDC capabilities to ensure that a container can be deployed using the desired software components in a suitable software-defined environment.

Future CM tools must also be able to handle the software-driven aspects of such an environment, in which containers usually exist for only minutes or even seconds. This puts a particular emphasis on [container orchestration tools for configuration management](#).

➤ Further reading

In this e-guide

-  What is configuration management? A comprehensive guide p. 2
-  How does configuration management work? p. 3
-  History of configuration management p. 5
-  Key benefits of configuration management p. 7
-  Risks of configuration management p. 10
-  Change and asset management p. 11
-  Software configuration management p. 13
-  Configuration management in DevOps p. 15
-  Configuration management tools p. 19
-  How to get started with configuration management implementation p. 21
-  The future of configuration management p. 23
-  Further reading p. 25

About SearchITOperations

SearchITOperations.com covers the latest industry news, technologies and trends that impact today's IT professionals tasked with designing and running applications created using DevOps, microservices and cloud-native paradigms.

As trends like hybrid cloud, containers, DevOps and more continue to infiltrate data center infrastructures, IT operations pros are facing a new set of responsibilities, such as integrating next-gen apps into their environments, running and supporting these apps, and collaborating with Agile/DevOps teams – that's where the expert content on SearchITOperations.com come into play.

Each of the contributors and editorial staff members at SearchITOperations.com, come with years of experience in the IT industry and have been hand-picked to produce the most objective, insightful and valuable IT operations-related content for our audience.

For further reading, visit us at
<http://SearchITOperations.com/>

Images; Fotolia

©2020 TechTarget. No part of this publication may be transmitted or reproduced in any form or by any means without written permission from the publisher.