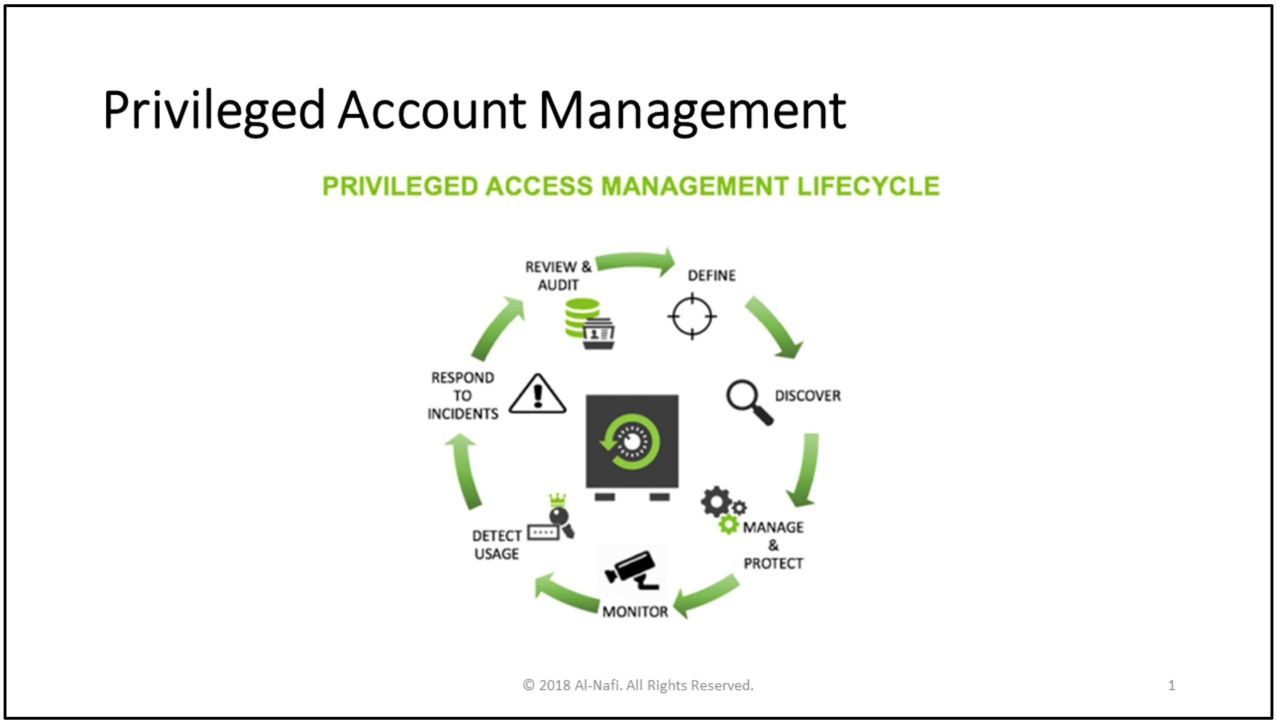


Privileged Account Management

PRIVILEGED ACCESS MANAGEMENT LIFECYCLE

Privileged accounts are those with permissions beyond that of normal users, such as managers and administrators. Because those permissions lend the privileged user more capability to cause potential harm to the organization, privileged accounts require additional protections. Typical measures used for attenuating elevated risks from privileged accounts include the following:

More extensive and detailed logging than regular user accounts. The record of privileged actions is vitally important, as both a deterrent (for privileged account holders that might be tempted to engage in untoward activity) and an administrative control (the logs can be audited and reviewed to detect and respond to malicious activity).

More advanced access control than regular user accounts. Password complexity requirements should be higher for privileged accounts than regular accounts, and
refresh rates should be more frequent (if regular users are required, for instance, to change passwords every 90 days, privileged account holders

might have to change them every 30). Privileged account access might also entail multifactor authentication, or other measures more stringent than regular log-on tasks.

Temporary access. Privileged accounts should necessarily be limited in duration; privileged users should only have access to systems/data for which they have clear need-to-know and only for the duration of the project/task for which that access
is necessary.

Deeper trust verification than regular users. Privileged account holders should be subject to more detailed background checks, stricter nondisclosure agreements, and acceptable use policies and be willing to be subject to financial investigation. l Greater audit of privileged accounts. Privileged account activity should be monitored and audited at a greater rate and extent than regular usage.

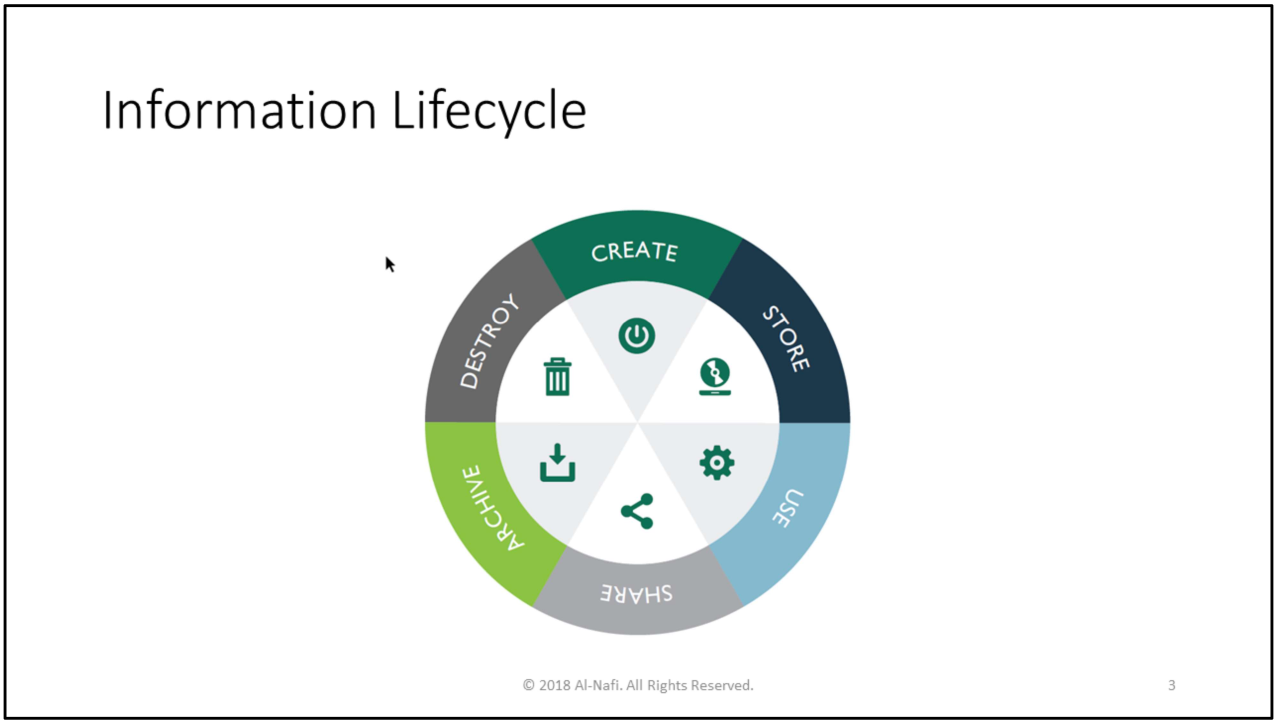
Job Rotation

The organization can implement the practice of job rotation, where all employees change roles and tasks on a regular basis. This improves the overall security of the organization in a number of ways:

An employee engaged in wrongdoing in a specific position may be found out when the replacement takes over that position after rotation.

The organization will have a staff that has no single point of failure; every person on a team will know how to perform all the functions of that team (to greater or lesser extent). This can be crucial for business continuity and disaster recovery actions.

This often improves morale, which fosters trust among employees; employees like having an increased skillset and marketability even if they don’t plan to leave the organization, and different tasks are intriguing and interesting and stave off boredom.



Information Lifecycle

3

Read this document from end to end
https://www2.deloitte.com/content/dam/Deloitte/us/Documents/finance/us-advisory-information-lifecycle-management.pdf

The data lifecycle stages can be described as the following:

Create: The moment the data is created or acquired by the organization.
Store: Near-time storage for further utilization; this takes place almost simultaneously with creation of the data.

Use: Any processing of the data by the organization.

Share: Dissemination of the data typically considered outside the organization (internal “sharing” would most often be considered “Use”); this can include sale of the data, publication, and so forth.

Archive: The data is moved from the operational environment to long-term storage; it is still available for irregular purposes (disaster recovery, for instance, or possibly to replace operational data that was accidentally deleted) but is no longer used on a regular basis.

Destroy: Data is permanently removed from the organization with no way to recover it.

The organization's security program should be sufficient to protect the data throughout all phases of the lifecycle with proper security controls for each phase.

Service-Level Agreements (SLAs)

© 2018 Al-Nafi. All Rights Reserved.

4

Service Level Agreement (SLA) Definition - What does *Service Level Agreement (SLA)* mean?

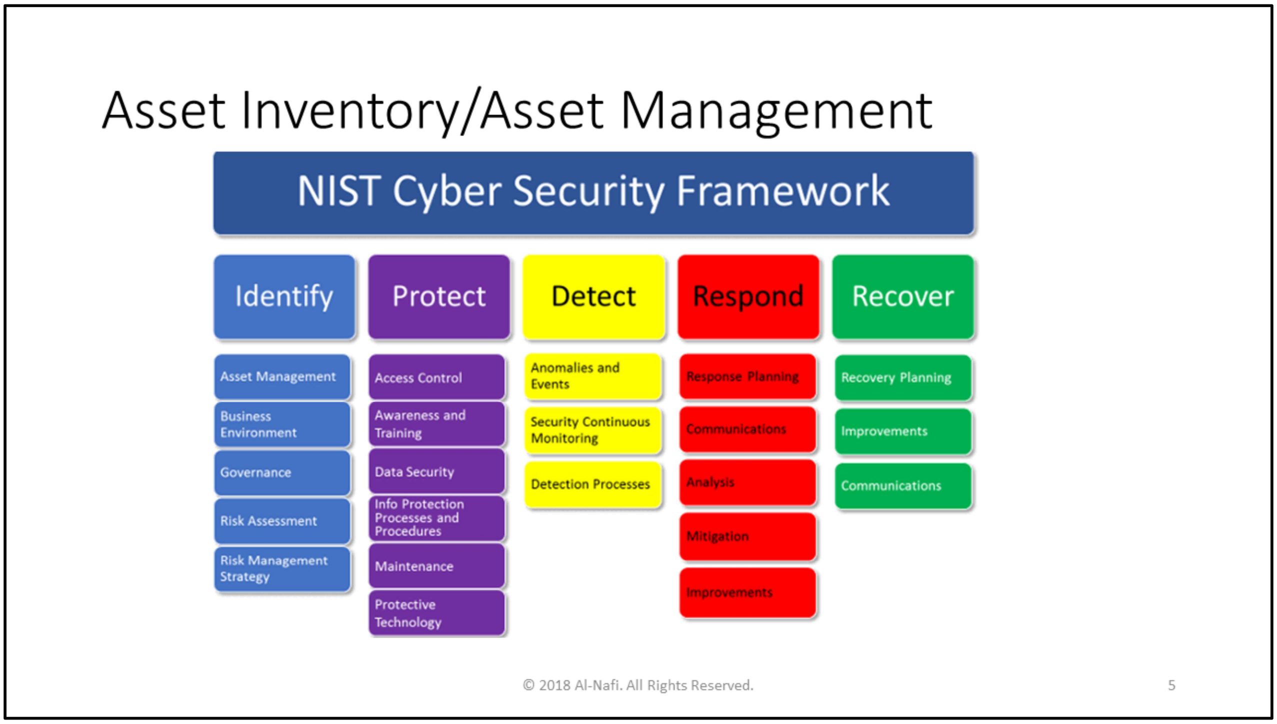
A Service Level Agreement (SLA) is the service contract component between a service provider and customer. A SLA provides specific and measurable aspects related to service offerings. For example, SLAs are often included in signed agreements between Internet service providers (ISP) and customers. SLA is also known as an operating level agreement (OLA) when used in an organization without an established or formal provider-customer relationship.

Adopted in the late 1980s, SLAs are currently used by most industries and markets. By nature, SLAs define service output but defer methodology to the service provider's discretion. Specific metrics vary by industry and SLA purpose.

SLAs features include:

- Specific details and scope of provided services, including priorities, responsibilities and guarantees

- Specific, expected and measurable services at minimum or target levels
- Informal or legally binding
- Descriptive tracking and reporting guidelines
- Detailed problem management procedures
- Detailed fees and expenses
- Customer duties and responsibilities
- Disaster recovery procedures
- Agreement termination clauses
- In outsourcing, a customer transfers partial business responsibilities to an external service provider. The SLA serves as an efficient contracting tool for current and continuous provider-customer work phases.



Asset Inventory/Asset Management

NIST Cyber Security Framework

Identify	Protect	Detect	Respond	Recover
Asset Management	Access Control	Anomalies and Events	Response Planning	Recovery Planning
Business Environment	Awareness and Training	Security Continuous Monitoring	Communications	Improvements
Governance	Data Security	Detection Processes	Analysis	Communications
Risk Assessment	Info Protection Processes and Procedures		Mitigation	
Risk Management Strategy	Maintenance		Improvements	
	Protective Technology			

Cybercrime is a growing epidemic that affects businesses of all sizes. Organisations have a responsibility to protect the data of their employees and customers. So they are investing in expensive hardware and software solutions. Yet businesses don’t realize that without effective management of those solutions, every component they add to their IT inventory becomes a new point of vulnerability. Cybercriminals can exploit unaccounted and out-of-date hardware and software to hack systems. So companies need to put effective IT asset management solutions in place.

What IT Asset Management (ITAM) Entails

IT managers have to keep track of their IT inventory. They have to deal with contracts, licenses, updates, and regulatory compliance issues. The use of the cloud and mobile devices are adding new layers of complexity. In the early days, managers could get away with using spreadsheets to keep track of their IT assets. Today most sophisticated operations use some form of IT inventory management software. These tools are better suited to deal with various aspects of IT asset management:

Hardware Asset Management: IT departments have been dealing with servers and workstations for a long time. But that doesn't mean that it has gotten any easier. A good ITAM practice requires that hardware is properly tagged and tracked throughout its lifecycle. The firmware of each hardware needs to be updated regularly. A good IT inventory management software has the provisions to handle the complexity of dealing with various aspects of hardware management.

Software Asset Management: Software provides a different set of challenges. IT departments have to prevent unauthorized software installations. They have to ensure security updates are regularly applied to installed applications and access management rules are followed properly. Good ITAM tools can keep track of software updates, license expirations, and compliance requirements. Regulatory audits are easier with software asset management.

Cloud Asset Management: Cloud-based services like SaaS, IaaS and PaaS are relatively new developments. So IT departments are still trying to figure out how to address various issues. In a pre-cloud environment, teams had total control over the IT inventory. But cloud environments use the shared responsibility model. Most ITAM tools are still not highly evolved for cloud asset management. So IT teams need to pay special attention in this area.

End-User Mobile Device Management: More companies are adopting bring-your-own-device (BYOB) policies. Even though its great for productivity, its a nightmare for implementing security. Tracking and monitoring BYOB devices through IT inventory management is a high priority for IT departments.

Why ITAM is Crucial for Effective Cybersecurity

For any modern organisation, it's not possible to create a robust cybersecurity program without having an efficient ITAM solution. There are just too many tools and services to keep track of.

For example, a single employee might have a PC, a mobile phone, and a tablet. In addition, the employee might have access to various servers and cloud applications. If cybercriminals can obtain even one password to any of these endpoints, they can often use that password to hack into other systems to gain more valuable information.

Also, cybercriminals can launch sophisticated phishing attacks, exploit software vulnerabilities or steal employee devices. IT teams need to fight battles on all fronts by keeping software and hardware up-to-date and having the capability to shut down stolen devices. Recent attacks in the UK shows cybercriminals are taking advantage of all these vulnerabilities.

British Airways Hack: Financial information of around 380,000 British Airways passengers were hacked during a 15-day breach in August 2018. Initially, British Airways didn't know how the hackers got access to the data as there wasn't any internal breach. Later security experts discovered that the scripts for its baggage claim information page were changed just before the hack started. The cybercriminals exploited the weaknesses of those scripts to intercept customer information. This shows an important reason for having ITAM solution. There is no information available about how BA managed its IT inventory in this case. But good ITAM solution would make finding vulnerabilities like this easier for security experts. Experts would be able to discover problems faster using ITAM historical data. Without proper ITAM, the same task will take significantly longer or even make the problem untrackable. It will increase the chances of future attacks.

NHS WannaCry Attack: The WannaCry ransomware attack of UK's National Health Services (NHS) caused canceling of 19,500 medical appointments, locking of 600 computers at GP surgeries and put 5 emergency centers out of service. The damage could have been worse if a security researcher hadn't accidentally discovered the kill-switch to the ransomware. But this attack could have been prevented in the first place through IT asset management. If NHS had updated their Windows operating system properly, the WannaCry could not have caused this havoc.

Establishing a Cyber Resilient Business Using IT Asset Management

IT asset management will not solve cybersecurity problems automatically. Businesses need to design and implement their IT inventory management software with cybersecurity assessment in mind.
However, cybersecurity-aware ITAM solutions will help your business in multiple ways. Here are some of the benefits:

Visibility and Transparency

ITAM solutions designed with cybersecurity objectives will help you find security risks faster. If you have a configuration management database (CMDB) for your IT assets, you can easily pinpoint when a problem happens. With regulations like GDPR, this becomes more important as you are legally required to report your security breaches.

Early Security Threat Detection

Hardware asset management and software asset management tools keep historical records or logs of various information. This information is a great resource to recognize irregularities or anomalies. This data can help your business early detect cyber attacks and take preventive measures.

Data Traceability

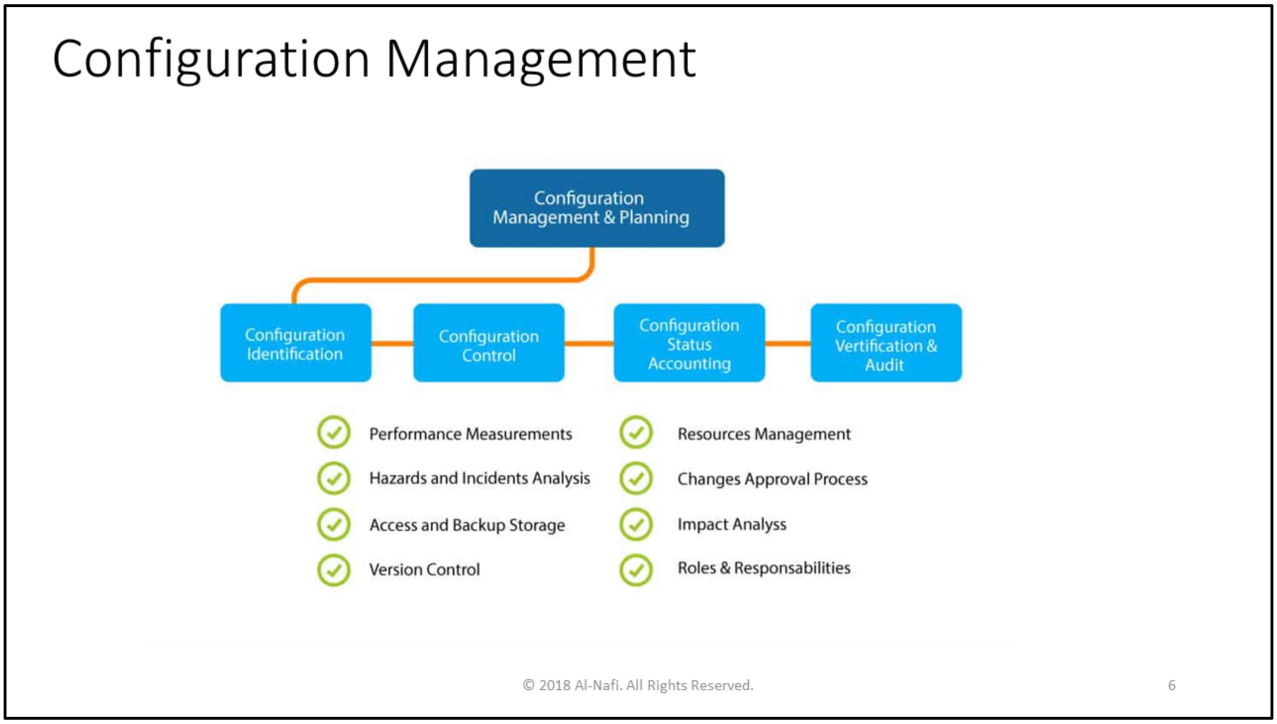
Data is the most valuable resource for businesses in the information age. Your ITAM solution gives you the ability to organize and align the data from your employees, your customers, and your infrastructure. So you'll have more control. It's an important tool for tracking and securing data.

Cost Optimisation

Cybersecurity is expensive. Most companies stop tracking their hardware or updating their software due to the associated costs. Initially, an IT inventory management solution might take resources to set up. But it will save you time and money in the long-run. It will make tracking and updating hardware and software assets easier and more efficient.

In Conclusion

No solution can stop all cyber attacks. But an ITAM solution can help your organisation build the necessary security strategies to improve your chances of preventing an attack. And a robust ITAM solution can help your business stay safer.



Configuration Management

Configuration Management & Planning

Configuration Identification

Configuration Control

Configuration Status Accounting

Configuration Vertification & Audit

- ✓ Performance Measurements
- ✓ Hazards and Incidents Analysis
- ✓ Access and Backup Storage
- ✓ Version Control
- ✓ Resources Management
- ✓ Changes Approval Process
- ✓ Impact Analysys
- ✓ Roles & Responsabilities

6

What Is Configuration Management?

Here's my definition of configuration management: it's the discipline of ensuring that all software and hardware assets which a company owns are known and tracked at all times—any future changes to these assets are known and tracked. You can think of configuration management like an always up-to-date inventory for your technology assets, a single source of truth.

Configuration Management & Planning With that defined, let's talk about how it works in practice. Configuration management usually spans a few areas. It often relates to different ideas, like creating "software pipelines" to build and test our software artifacts. Or it might relate to writing "infrastructure-as-code" to capture in code the current state of our infrastructure. And it could mean incorporating configuration management tools such as Chef, Puppet, and Ansible to store the current state of our servers.

Where Did Configuration Management Originate?

When I first started learning about configuration management, I found the concept *super* confusing. However, it turns out that there are reasons for the confusion. But to understand why, we need to look at some history. We (the Software Industry) Stole the Idea of “Configuration Management” The idea of configuration management comes from other institutions, such as the military. We took those ideas and retrofitted them into a software context.

How We Make Software Has Changed Over Time
Configuration management was traditionally a purely manual task, completed by a systems administrator. The role was a lot of manual work involving carefully documenting the state of the system. But the industry has changed completely. These changes came from the popularity of DevOps, increases in cloud computing, and new automation tooling. Now that we’ve set the scene, we can dive into the details of configuration management. So let’s get to it!

What the World Looks Like With Configuration Management
Before we explore different tools for configuration management, we need to know what end results we’ll receive for our efforts.
What are the outcomes of well-implemented configuration management?
Let’s cover the benefits.

Benefit 1: Disaster Recovery
If the worst does happen, configuration management ensures that our assets are easily recoverable. The same applies to rollbacks. Configuration management makes it so that when we’ve put out bad code, we can go back to the state of our software *before* the change.

Benefit 2: Uptime and Site Reliability
The term “site reliability” refers to how often your service is up. I’ve worked at companies where each second of downtime would cost thousands—often tens or even hundreds of thousands. Eek!

A frequent cause of downtime is bad deployments, which can be caused by differences in running production servers to test servers. With our configuration managed properly, our test environments can mimic production, so there’s less chance of a nasty surprise.

Benefit 3: Easier Scaling
Provisioning is the act of adding more resources (usually servers) to our running application. Configuration management ensures that we know what a good state of our service is. That way, when we want to increase the number of servers that we run, it’s simply a case of clicking a button or running a script. The goal is really to make provisioning a non-event.
These are just some of the benefits of configuration management. But there are some other ones, too. You’ll experience faster onboarding of new team members, easier collaboration between teams, and extended software lifecycle of products/assets, among other benefits.

The World Without Configuration Management
Sometimes it’s easier to grasp a concept by understanding its antithesis. What does trouble look like for configuration management, and what are we trying to avoid? Let’s take a look. A developer implementing a feature will commonly install a few bits of software and deploy code. If things are sloppy, this developer probably makes the team and manager aware of the intention to come back later to clean it all up—that it’s simply a demonstration and will be rewritten soon.

But then the deadline starts pressing, and the task of going back through and rewriting the installation steps as a script gets pushed lower and lower in priority. Before we know it, several years have passed, and a new developer gets put on the project. That developer is now left to pick up the pieces, trying to understand what happened. It’s quite likely they aren’t even going to touch the configuration of the server. Who knows what it would do!
The above situation is precisely what configuration management helps you avoid. We don’t want to be left in the dark as a result of developers setting up software without proper documentation/traceability. Rather, we want to know the answers to questions like

- What services are we running?
- What state are those services in?
- How did they get to their current state?
- What was the purpose for the changes?

Configuration management can tell us these answers.
That hopefully paints a clearer picture of the problems that configuration

management is trying to solve. How Configuration Management Fits in With DevOps, Continuous Delivery, and More…

Hopefully by now you're starting to get the hang of what configuration management is and what it aims to do. Before we go on to discuss tooling, I'd like to take a moment to address how configuration management fits in with other software development concepts like agile, DevOps, continuous integration, continuous delivery, and Docker so that you can understand how these concepts fit in with the ideas of configuration management.

Is Configuration Management Compatible With Agile?
Yes. Agile software, by definition, reflects the desire to make changes to our software faster so that we can respond to market demands. Configuration management helps us to safely manage our changes and keep velocity high.

How Does Configuration Management Fit With DevOps?
DevOps is the extension of agile practices across both the development and operations departments. In fact, DevOps seeks to unify the goals of both departments. At some companies, the development department seeks change while the operations department seeks stability. But companies that embrace DevOps want both stability of their deployed assets and frequency of change. However, achieving this outcome requires cultural change.

Like agile, configuration management gives teams the confidence to move quickly with their changes. Under agile practices, the company gives configuration management responsibilities to the development teams, empowering them to provision, configure, and manage their own infrastructure. You build it, you run it.

Where Do Pipelines Fit Into Configuration Management"?
Software pipelines are the steps (or "value stream," which we can create with tools like Plutora) that we usually automate, taking code from commit to production. Pipelines usually involve steps such as linting code, unit testing code, integration testing code, and creating artifacts. A software pipeline therefore is a form of configuration management. When we build software with tools like Docker, we codify our build instructions into our Dockerfile. This allows us to better understand the dependencies of our artifacts.

Is Infrastructure-as-Code Configuration Management?

Infrastructure-as-code (or IaC for short) is the practice of ensuring all provisioned infrastructure is done so through code. The purpose of IaC is to have a written record of which services exist, where they are located, and under what circumstance. Configuration management might choose to leverage aspects of IaC in order to achieve the full understanding of all the technology assets a company owns.

Is Continuous Integration/Delivery Configuration Management?

Continuous delivery is the process of ensuring that software is always in a releasable state. You can achieve this through heavy automation and testing. Continuous integration is the process of bringing separate software artifacts together into a single location on a frequent basis, for the purposes of verifying that the code integrates properly. Continuous integration tools, which are typically servers that run automation-testing suites, act as a form of configuration management by providing visibility into the steps required to set up and configure a given software artifact.

That should clear up some of your lingering questions about how configuration management fits with some practices or ideas that you might be using or are familiar with. Any discussion of configuration management would be incomplete, however, without a discussion about tooling. So, let's take a peek at the different tools we have at our disposal for implementing configuration management.

The Importance of Declarative Style in Configuration Management Tools

Next up, we're going to discuss configuration management tools. But before we get to that, I need to quickly discuss a concept to consider when comparing tools. And the concept is declarative style. You'll hear about this terminology a lot if you go out and start looking into different configuration management tools. So, it makes sense to have a firm grasp of what declarative style is, why it's important, and why so many people are talking about it.

So, what do we mean by declarative style?
And why is declarative style so important for configuration management?

What Do We Mean by Declarative Style?

When it comes to software, having a declarative style means telling your software the end result you want and then letting the software do the work in figuring out the way to get there. The opposite of the declarative style would be a procedural style, where instead of giving an end state, you give instructions on how to get there. The problem with instructions is that they're dependent on the starting state.

You can think of it like this: declarative versus procedural is the difference between giving a friend your home address and giving them step-by-step instructions to get to your house from where they are. The problem with giving step-by-step instructions is that it assumes you know where the friend is starting, and it doesn't allow for things to go wrong. It's hard to replay steps when you're in a bad state (i.e., lost!).

Why Is Declarative Style Important for Configuration Management?

By now, you're probably thinking that declarative style sounds interesting. But why is it important?

Declarative style is important because configuration management is all about *knowing the current state of your applications*. So when we use configuration management tools, it's desirable to use a declarative style and specify the end result that we want, not the steps to get there. This means we always know what end state we're trying to achieve and how that's changed over time. That's instead of trying to work out when instructions were run and dealing with the complexities that may arise if certain instructions have failed.

What Are Configuration Management Tools?

There are many different tools for configuration management. In fact, it can get confusing, as there are tools that support configuration management without explicitly being configuration management tools.

For instance, Docker neatly packages up steps needed to set up and run an application (in a Dockerfile). However, people don't often consider Docker a configuration management tool.

To make things clearer, let's divide up the common tools that might fall under or relate to configuration management:

Configuration Management Tools

These are the tools you see typically associated with configuration management. Tools like Chef, Ansible, and Puppet provide ways to codify steps that we require in order to bring an asset in line with a current definition of how that asset should look. For instance, you might create an Ansible playbook that ensures that all of our X servers have Y installed within them.

Infrastructure-as-Code Tools

Often also called provisioning tools, IaC tools include CloudFormation and Terraform. If our configuration management tools include the setup we need on our assets, our provisioning tools are how we get those assets. It's this blurred line that explains why we need to bring these tools into our discussion of configuration management. And many consider it an anti-pattern to use configuration management tools for provisioning.

Pipeline Tools

We talked briefly about software delivery pipelines, but implementing them requires tooling. Popular technologies include Jenkins, CircleCI, and GitLab CI. By using tools to codify our build process, we make it easy for other developers to understand how our artifacts are modified and created, which is a form of configuration management.

Source Control Tools

Source control tools include GitHub, SVN, GitLab, and Bitbucket. While we need to codify our automation in scripts, if we don't appropriately track the history of our changes, then we aren't really achieving configuration management.

We're now nearing the end of our introduction to configuration management. We've covered what configuration management is, we know the benefits, and we're now up to date on the latest tools. However, all of this information can be a little overwhelming if you're asking the simple question of "Where should I start?"

Let's break it all down so that you can start your journey into configuration management.

How Can I Get Started With Configuration Management?

Where to start? Do you begin by researching tools? Implementing some automation? Auditing your existing servers? Talking to others in your company? Where you start with anything always depends on where you currently are. That said, only you are aware of your current situation and the limitations and resources available. Below are three different places you can begin your journey to effective configuration management:

Audit your software/hardware—What software do you currently have? What's the state of it? Is it well documented? Are the setup and run instructions known for the software?

Perform a tools assessment—Do an assessment of what tools exist on the market for configuration management. The ones I listed above are a good start. Identify which tools could help you solve some of your configuration management problems.

Learn about best practices—Successfully implementing configuration management isn't a one-and-done task. It takes time and work to continually ensure that all new software is appropriately audited and tracked. So you might want to look into some different key concepts, such as IaC and build and release pipelines.

It's Time For Everything-as-Code!

And that's all! Hopefully that helps to clear things up for you about configuration management. It's all about keeping track of the current state of your software and infrastructure.

There are many ways to implement configuration management, and there are lots of different tools and processes. So when it comes to strategy, be sure to take your time assessing options and understanding how you want your configuration management processes to work.

It will all be worth it in the end, though. Get your configuration management right and your teams will be safer, more productive, and faster to make changes!

Good luck—and from now on, audit, track, and write everything-as-code!