AUTOMATED CONFIGURATION MANAGEMENT

Why it Matters and How to Get Started

puppet labs

Configuration management: A definition

Configuration management is a critical cornerstone of IT automation, providing tools that allow you to centrally manage the packages, configuration files, process state, firewall rules and other settings that equip servers to do their assigned jobs. Without automated configuration management, you do all these tasks manually, or with handwritten scripts. It's time-consuming, and prone to human error.

Configuration management is also about making any changes to the system in an organized way so that your servers are modified deliberately and correctly, while accounting for relationships between system components.

People often talk about configuration management as if it referred only to compute servers. But it's also about managing network devices, storage, and the applications you're running. So really, you can think of configuration management in IT as making sure all the machines in your data center — and in the cloud, too — are equipped to do the jobs they're supposed to do, that they're actually doing those jobs, and that the overall system is functioning well to run the applications that serve the business.

PART ONE

Why you need automated configuration management

There's enough stress in your job without having to repeat the same tasks over and over again, continually put out fires, or hear for the hundred-thousandth time, "It worked on *my* machine." Automation can eliminate (or nearly eliminate) these things from your life, and make your IT environment more stable and predictable.

Not automating configuration management causes pain

Sysadmins have been managing system configuration for decades with tools and processes such as shell scripts, golden images, kickstart files, and server setup checklists in spreadsheets. While these methods can give you a repeatable, standardized way to build systems, they don't provide you a way to keep your system in the desired state. You can get to the desired state with a great setup script, for example, but that script won't tell you when configuration drift happens due to small manual changes or automated updates. Nor can that script reset servers to the correct state — it takes human intervention to do that.

There's also that little matter of being asked to do a lot more with the same resources. You're probably familiar with this: business managers informing you that competitors and other companies are launching new features faster, and asking you to do the same — or even better. You know that cloud services and internal innovations, such as continuous delivery, could make the difference. But if you're managing your system with scripts, you also know that an expanding infrastructure will further expose the problems and limitations you're already experiencing:

- **Manual errors.** If you're going from one server to the next, installing the same package on all of them, you'll do it right nine times out of 10 … and the tenth time is what causes problems later.
- **Custom scripts can be fragile.** Something as simple as a security update can break a script.
- **Custom scripts aren't accessible to everyone.** Writing in Perl? Sed and AWK? Python? Colleagues and new teammates may not be able to readily understand your work, which makes collaboration harder.
- **You can't scale.** When the business wants to move to the cloud, or asks for faster provisioning of development and test environments, it's tough to add all these new services without automation. You have only so many hours in the day; at a certain point, you (and your team) simply can't do it all manually.
- **You don't always know what went wrong.** Something broke; it takes forever to find what broke, and maybe you can't even locate it. Now you don't know how to prevent the breakdown from happening again, and you've spent way too much time trying to figure it out — valuable time that could have been spent on something more strategic.
- **You're not getting to other things that matter.** Are your security protocols so complex that people are bypassing them? Maybe it takes far too long to run payroll, or your servers are running an OS version that won't support a new software package you are supposed to install next week. When you spend too much time on routine system management, you can't prepare and plan sufficiently to meet business requirements.

Automated configuration management eliminates a lot of manual work, and creates greater dependability and predictability. You specify the configurations, and the configuration management (CM) system continually brings servers back to the state you defined. Updates, status checks and remediation are all done much faster, and if the software includes robust reporting, failed changes and incorrect configurations will be pinpointed for you. If, for example, a junior sysadmin installs a package outside of normal channels, or edits a configuration file, these changes will be detected, logged, and reverted. In the case of a CM solution like Puppet, you can also choose to run in “no-op” mode if you just want an audit and report on the system, and you don't want changes made automatically.

Another problem with the manual-change and script-based approach is the lack of a single source of truth. You have scripts that were used to build the server, documentation and change logs, and the actual configuration of the server at any given moment. These three things aren't automatically synced, and they fall out of sync over time. With automated configuration management, you have a single source that's always in sync with the actual state of your systems.

You can quickly and automatically verify that your test and staging environments are configured the same as your production environment. That gives both sysadmins and developers the confidence that code under development will actually run once in production, eliminating the dreaded, "Well, it ran on *my* machine." That's important not only for companies whose business is creating software, but for any company running customizable software — for example, accounting packages, customer relationship management systems, e-commerce platforms, compliance software, security software and more.

Best of all, automation makes you more productive; lets the people who are dependent on you be more productive; and gives you the time to uplevel your job and your skills. Spend less time on manual configuration and putting out the inevitable fires, and you'll have more time to do things like:

- Properly evaluate software other areas of your company want to use, and give well-argued advice on which package to pick. After all, you're going to end up supporting the thing.
- Test backups.
- Revisit the disaster recovery plan and see if it's still current and would really work.
- Fix all the things the last person in your job didn't do very well.
- Learn something new and interesting.

Everyone benefits from automated configuration management

Automated configuration management makes things better for everyone in your company, whether directly or indirectly. After all, your entire company depends on a properly functioning IT system — from developers creating new products or sales interfaces; to sales tracking customer contacts; to marketing creating new leads; to accounting making it all make sense, not to mention getting everyone paid on time.

Let's look at all the benefits of automated configuration management across the organization:

Sysadmins get:

- visibility into the state of the overall system, and each component of it
- faster identification of problems, and faster remediation
- efficiency in performing regular maintenance
- documentation, so fixes and updates can be referenced and repeated
- an end to 3:00 am emergency calls
- time freed up to think about, and do, more strategic work

Quality assurance and test engineers get:

- the confidence that they're testing code in an environment that's identical to where the code was created and where it will finally run

IT managers get:

- visibility into what the IT team is doing: maintenance, remediation, provisioning
- faster time to recovery, so greater satisfaction for internal and external customers more efficient, predictable operations
- better ability to plan
- better ability to report to other teams in the company

The entire company gets:

- more stable and dependable systems and software to work on
- faster service from IT when things need to be installed or fixed
- the ability to expand IT services while containing costs
- the confidence to make changes more quickly and learn from market response

Customers (the people who pay us!) get:

- better products, better services, faster innovation and better response to their needs

Automated configuration management helps you improve and innovate

Automated configuration management is a cornerstone for many things that can help your company move faster, satisfy customers better and improve efficiency.

- **Continuous integration and continuous delivery.** If you want to make changes to software and your IT system more frequently, with less pain, you need to be using a version control system, which is integral to achieving continuous delivery. Automated configuration management is an essential component of this process for deploying more frequently, dependably.
- **Cloud computing.** Most companies using cloud services also have a physical data center. Automated configuration management is critical to making sure that both cloud and physical machines are configured appropriately so you can "burst" to the cloud as needed, and allow developers to self-service environments that closely resemble the production environment.
- **Infrastructure as code.** If you're going to manage your entire infrastructure through a version control system and automated workflows, you must automate configuration first.
- **DevOps.** Getting sysadmins, developers and test engineers onto the same tools is a core piece of the strong collaboration and faster momentum that people seek from DevOps. Configuration management is one of the key components of the DevOps toolchain.

Now you have a good idea of why you need configuration management software, and if you already knew you wanted it, you have plenty of arguments in its favor to go and make your case. So let's get started!

PART TWO

How to get started with configuration management

Here's the secret: start small.

Automate one small thing

Think about the things you do over and over again, or that are frustratingly error-prone. You'll learn the most about how automated configuration management can work for you if you start by choosing **one small thing**, in an **environment you control,** that will make a **real impact on your life** because it will save time, reduce errors, please other team members — or all three.

Many sysadmins start by automating management of something like NTP, DNS, SSH, firewalls, or users and groups — all things that are completely routine, and that suck up a lot of time. But these aren't the only things you could start with. Maybe your biggest pain point is application deployment, or patching and updating, or locating and remediating configuration drift.

Making a careful choice of what to start automating will give you a quick and clear read on what automation can do for you and your team — and will probably tell you what to do next.

Evaluate for capabilities and integrations you actually need

If you're going to use a configuration management (CM) solution, you'll want one that works for *your* needs. So make sure the things you decide to automate first help you choose a technology that will work for you as the business grows and scales. Look for a solution that:

- **Supports basic configuration management and more complex operations.** Many sysadmins start by automating basic server configuration, then moving on to patching and updating. Then they enter more complex territory, such as configuring development and test environments to align with the production environment, or configuring cloud instances. At this point, you'll be looking for more capabilities from your CM solution, and support for a wider range of technologies. Identify your most basic configuration needs, and your more complex situations, to help you better evaluate solutions, and choose one that can support all configuration management needs.

- **Supports today's use cases *and* your technology roadmap.** Today's business isn't next year's business. Maybe your databases will expand exponentially if your new business line succeeds; maybe you'll be adding call centers in multiple geographies. Whatever your needs will be, you need to think about them now, so you don't sink time into learning a system that can't handle a larger, more complex business than the one your IT environment serves today.
- **Offers pre-built solutions for the key systems you need to manage.** You want automation so you can increase productivity and agility — so make sure you're looking at a solution that helps you manage the most fundamental elements of your IT environment. For example, if you use PostgreSQL or MySQL, you don't want a solution that makes you write lots of custom code to get started. You'll be getting the full advantage of automation if you choose a package that has pre-built solutions for these, and for the other technologies that are core to your business. Starting with something readymade will save you time, and you'll have the comfort of knowing the solution has already been tested by other users.
- **Fits into your automation tool chain.** A complete IT automation solution is comprised of multiple tools that do things other than configuration management: monitoring, continuous integration, version control, ticket tracking, code review and more. Make sure the configuration management technology you select integrates with the other automation tools you rely on now, or plan to adopt.

Choose a tool that supports agility and collaboration

You don't work on an island. Other teams (development, for example) also need to work with the IT environment, or at the very least, understand how it works. The following attributes will turn your CM system into a tool for better collaboration and faster deployments.

Declarative model

"Infrastructure as code" has become a popular phrase, but all approaches to managing infrastructure as code are not equal. The most powerful and efficient solutions use a declarative model: you define the desired state, and the system makes it happen — across environments, different operating systems and different devices. The declarative approach lets you focus on the outcome you want, not on the procedural steps and dependencies of getting there.

Easy-to-read syntax

Your team counts on each other to keep systems running smoothly, so you need a shared language for easy communication. Many systems rely on users knowing programming languages like Ruby. This can work if everyone on your team is proficient in the same language, but if not, it's difficult to analyze the impact of a given chunk of code. Choosing a CM tool that works with simple, readable code means everyone can quickly understand what the code will do, and how it could affect other things in the environment.

Centralized management

Have you ever worked in a company that acquired another company, and its IT environment? Yeah, that was bad. Centralized configuration management lets you manage all machines — including those set up in a totally different context — in compliance with your company's policies, including security policies or those imposed by regulatory bodies. A centralized CM system keeps all facts about machines in a defined, secure location, administers security certificates, and applies changes in a uniform manner.

With centralized management, you can make a change in one place and then easily push it to multiple machines. No more walking from rack to rack, configuring one machine at a time.

Simulation

Whether you're spinning up new servers or remediating configuration drift, simulation makes it possible to test changes before you enforce them. Surprisingly, many CM tools do not let you accurately simulate the desired state. Some tools are procedural execution tools and don't allow for testing, and others simply execute scripts with no way of determining the current state of a node or its desired state.

Enforcement and containment

Before choosing a tool, investigate its enforcement process, and what happens when the tool finds a machine that isn't configured correctly. You don't want a failure to halt the entire process of evaluating machines and enforcing correct configurations, and tools that execute commands in a serial fashion actually do stop the process when a machine fails. That means the failed machine is left in an incomplete state, and you (or someone else) has to go in and fix it by hand.

If you choose a declarative CM solution that contains failures, the evaluation and enforcement process continues to run, and then returns you a report.

Reporting

A solid CM solution gives you complete insight into your infrastructure, with details about which machines failed, and why. These insights make your decisions around remediation and redesign faster and more efficient, and are especially valuable when it comes to identifying security vulnerabilities.

Access to a community of experts and tested solutions

Luckily, you don't have to wade into automated configuration management alone. Others have solved these problems before you, so make good use of their work! Poke around on community forums and help sites for the tools you are evaluating, and check out how responsive and engaged the community is. You'll get some answers to your questions, but make sure you also look for tested pre-built solutions that will help you get up and running quickly.

Now let's *actually* get started

Ready to automate? Puppet Enterprise is an IT automation solution that gives you the power to easily automate repetitive tasks, quickly deploy critical applications, and proactively manage infrastructure, both on-premise and in the cloud.

Give Puppet Enterprise a try:

- The Puppet Labs Workshop lets you work through free online courses, each covering a fundamental element of Puppet.
- Download our Learning VM, which guides you through a series of fun quests to learn all the fundamentals of IT configuration management with Puppet Enterprise.

Questions?

Stop by Ask.PuppetLabs.com or contact our sales team.

Acknowledgements

Authors: Aliza Earnshaw, Tim Zonca

Editors: Aliza Earnshaw, Molly Niendorf

With thanks to Kent Bye, Michelle Carroll, Daniel Dreier, Isaac Eldridge, Christy McCreath, Reid Vandewiele, Chris Westphal