

Home » Resources » Blog » Puppet Vs. Chef: Key Capabilities, Use Cases + A Comparison Table

June 5, 2023

Puppet vs. Chef: Key Capabilities, Use Cases + A Comparison Table

ECOSYSTEMS & INTEGRATIONS

By [Stephen P. Potter](#)

Choosing an automation and configuration management solution like Puppet vs. Chef often feels like comparing apples to oranges (or apples to... differently shaped apples). From the outside, it's hard to tell which tool does what, let alone which will perform best in your infrastructure.

If you're researching Puppet vs. Chef, this blog will explain some of the commonalities between Puppet and Chef, what makes Puppet unique, and what to consider when researching Puppet vs. Chef for your infrastructure automation and configuration management needs.

See for Yourself What Makes Puppet Better Than Chef

You could keep reading for a head-to-head comparison... or you can see Puppet in action right away.

[GET A DEMO OF PUPPET](#)

Table of Contents

- [What are Chef and Puppet?](#)
- [What's the Difference Between Puppet vs. Chef?](#)
- [What's Unique to Puppet?](#)
- [What Do Puppet and Chef Have in Common?](#)
- [Is Puppet Better than Chef? Is Chef Better than Puppet?](#)
- [Puppet vs. Chef Comparison Table](#)
- [What Puppet Does Better Than Chef](#)
- [See Puppet in Action or Try It for Yourself](#)

What are Chef and Puppet?

Chef and Puppet are both configuration management platforms that aim to configure and manage servers and other IT infrastructure.

What's the Difference Between Puppet vs. Chef?

The main differences between Puppet and Chef include use cases, scalability, reporting, community support, and out-of-the-box features.

What's Unique to Puppet?

Puppet DSL's Automatic Relationship Management

Puppet's Domain-Specific Language (DSL) is a declarative language that describes the desired state of your system. That means you tell Puppet the configuration and dependencies you want to see, and Puppet will do everything it needs to get it to that desired state. Puppet is unique in its ability to do automatic relationship management and implicit resource ordering.

Built-In Reporting + Visibility

Puppet provides a built-in reporting feature that captures information about the configuration changes and the state of managed nodes. It can generate reports and visualizations that help in tracking the infrastructure state over time. Chef, on the other hand, relies on external tools and integrations for reporting and visualization.

With Puppet, Conditional Logic is Determined on the Server

The difference between Puppet and Chef is where the logic takes place. With Puppet, catalogs are compiled on the primary server (also known as a compiler server in a large-scale environment) and the unambiguous catalog (no logic or conditions) is sent to the agent for execution. With Puppet, the workload is on the server.

Try Puppet Now

If you're ready to get your hands dirty.

[FREE TRIAL](#)

Get a Live Demo

If you'd rather see Puppet in action.

[PRODUCT DEMO](#)

What Do Puppet and Chef Have in Common?

Puppet and Chef may be different tools, but they both operate in the same infrastructure management space. That means they have a lot in common, since their offerings have been shaped by the same trends and feedback over the years. Here are some of the similarities Puppet and Chef both share.

Puppet and Chef Can Both Be Used for Configuration Management

Puppet and Chef can both help you automate aspects of your infrastructure management, like machine provisioning (standing up a virtual machine, laying down the operating system, etc.) and enforcing compliance. They allow you to define system configurations, package installs, file management, and more to automate management of your infrastructure.

Chef and Puppet Both Support Cross-Platform Configurations

Puppet and Chef can manage configurations across multiple operating systems, including Windows, macOS, Linux, and Unix. Platform-independent abstractions in each make it possible to write configurations that work across different platforms, OSes, and environments.

We'll Show You Why Puppet is the Right Choice for Automating + Configuring Your Infrastructure.

[GET A DEMO OF PUPPET](#)

Puppet and Chef Both Use "Pull" Architecture

Puppet agents are installed on target nodes and communicate with a central Puppet server. In this agent-based architecture, Puppet agents pull updates from the Puppet server and apply them to the nodes. Chef uses a similar "pull" architecture.

Puppet and Chef Both Have Community and Ecosystem Support

Puppet and Chef both have communities of active users. These communities gather, share knowledge, and contribute content that improves the tool and extends its functionality into new and more convenient use cases.

- Puppet has modules**, which are files written in Puppet DSL that describe system configurations which Puppet can enforce. Modules are hosted on the Puppet Forge, which is home to more than 6,000 modules – including ones created by the community and many officially supported by Puppet.
- Chef has Cookbooks**, which, like modules, are files that describe how a system should be configured and the actions Chef needs to take to get it there. Cookbooks are hosted on the Chef Supermarket, a hub like the Puppet Forge.

Is Puppet Better than Chef? Is Chef Better than Puppet?

Puppet and Chef feature many similar capabilities and use cases. Puppet is known to be better for managing large-scale deployments across data centers and the cloud. Chef is widely used to manage smaller, less complex infrastructure.

Of course, your choice of infrastructure automation and configuration management tools comes down to what your infrastructure needs today, what your team can manage, and what you want to do with your infrastructure in the future.

For example, if your organization plans to add node capacity or diversify its IT infrastructure, you'll need an automation tool built for scalability across physical servers and public or private cloud services. You'll also need a tool that integrates smoothly with your current tech stack and platform.

To help you make a more informed choice between Puppet and Chef, take a look at the technical comparison table below.

Puppet vs. Chef Comparison Table

Platforms	Commercial: Linux, Windows, MacOS, AIX, Solaris Open Source: FreeBSD	Commercial: Linux, Windows, MacOS, AIX, Solaris, FreeBSD Open Source: Arista EOS, Virtuozzo, XCP-ng
Language	PuppetDSL for desired state and task-based capabilities	Ruby-based DSL
Architecture	Server/client OR client-less ("pull")	Server/client OR client-less ("pull")
Interface	Puppet Enterprise provides a GUI with visibility to events & config details	Chef Automate provides a GUI with visibility to events & config details
Setup	Built to scale with your automation needs	Chef can be scaled through integrated components
Community	A bustling dev community and thousands of modules on the Forge (including many supported by Puppet)	Chef's dev community is smaller, with about half as many modules on the Supermarket and very few supported modules
Free Trial	Puppet Enterprise's free trial allows you to automate 10 nodes for free as long as you want	Chef Automate offers a 60-day limited trial
Scalability	Designed to scale for enterprise automation	Chef can scale because load is forced onto the client – but internal adoption by teams is often slow
Visibility & Reporting	Robust visibility and reporting built into Puppet Enterprise (Impact Analysis available in CD4PE)	Accessible only with extensions
Management	Puppet DSL and some YAML	Ruby-based DSL, Full Ruby, YAML, InSpec
Cloud Availability	AWS, Azure, GCP + more	AWS, Azure, GCP + more
Communication	SSL	SSL, SSH, WinRM

What Puppet Does Better Than Chef

Your choice of an automation and configuration management tool will depend on your specific needs. That said, there are some differentiating factors that make Puppet the tool of choice for engineers, sysadmins, and business leaders. Here are a few ways Puppet outpaces Chef.

Puppet is a More Complete Compliance Management Tool Than Chef

Puppet's infrastructure-as-code (IaC) lets organizations codify their configurations in version control, which makes it easier to prove continuous compliance during an audit. Puppet Comply, an extension for Puppet Enterprise, features in-depth auditing against expert-defined [CIS Benchmarks](#) to track configurations and changes over time.

Puppet can also generate compliance reports and dashboard metrics so teams always know their compliance standing. Puppets uses policy as code to automatically remediate configuration drift against CIS and [DISA STIGs](#) security standards to keep systems in compliance with regulations and frameworks.

Puppet Has a Lighter Client Than Chef

As mentioned above, with Puppet, all of the compilation/logic takes place on the Puppet servers. That means the agent nodes have a lighter workload to complete, so there's less chance of a Puppet run affecting the processing on a heavily utilized node.

Puppet Can Reorder Resources Automatically

Implicit relationships (or automatic relationships) refers to Puppet's ability to reorder resources in a catalog based on known requirements. This means you can put things in a manifest out of order and in some/many cases, Puppet will reorder them to do the right thing.

For more on implicit relationships in Puppet, see Docs on [relationships and ordering](#) and [automatic relationships](#) >>

Puppet Has Been Around Longer Than Chef

Nothing proves the long-term value of a tool like time. Longevity matters for a number of reasons: More releases mean more features and better reliability. But Puppet has also used that time to leave a huge footprint, including a library of [DevOps reports](#) and a gigantic [Puppet community](#) of passionate contributors helping to make every release a little better.

Puppet Has More Automation Use Cases Than Chef

Puppet's use cases in automation make it the most versatile IT and infrastructure automation solution available. Puppet's automation use cases include [system configuration](#), [patch management](#), monitoring, source control, [secrets management](#), and more.

Chef, on the other hand, lacks key capabilities like patch management. In [The Forrester Wave™: Infrastructure Automation, Q1 2023](#), Forrester described the limited use cases of Progress Chef:

“Progress does not offer a native patch management solution; its capabilities for the rest of automation use cases are relatively weak. ... Progress's vision and roadmap plans are lackluster. It intends to expand use cases, build functionality for DevOps pros, and enable vertical capabilities — but this falls short of its competitors.”

The Forrester Wave™: Infrastructure Automation, Q1 2023

Puppet is Built to Grow with Your Business

[Scaling DevOps](#) is essential to helping your IT infrastructure meet the growing needs of your business – and it's one of Puppet's greatest strengths. Puppet's automation and configuration management capabilities go beyond provisioning, testing, and deployment – it includes reporting, remediation, enforcement, and more. With those tools under your control, you can scale infrastructure as needed with fewer headaches and less risk.

Chef, on the other hand, is known to be an impediment to scaling. Forrester reported that for Chef customers, “it took a long time to scale internal adoption and feel comfortable with advanced configuration and maintenance. They also reported that the documentation isn't straightforward, making it hard to get up to speed quickly.”

See Puppet in Action or Try It for Yourself

If you're evaluating infrastructure automation and configuration management tools like Puppet and Chef, you should absolutely do as much hands-on research as possible. That's why we offer a free demo of Puppet products, as well as a trial of Puppet Enterprise you can run on up to 10 nodes as long as you want – for free.

[GET A DEMO](#)

[TRY PUPPET FREE](#)

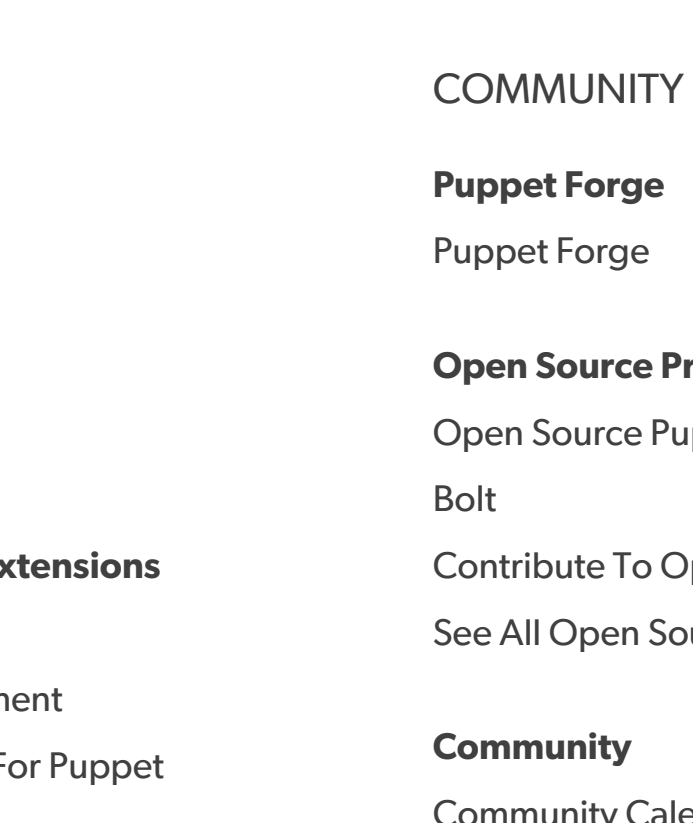


Stephen P. Potter

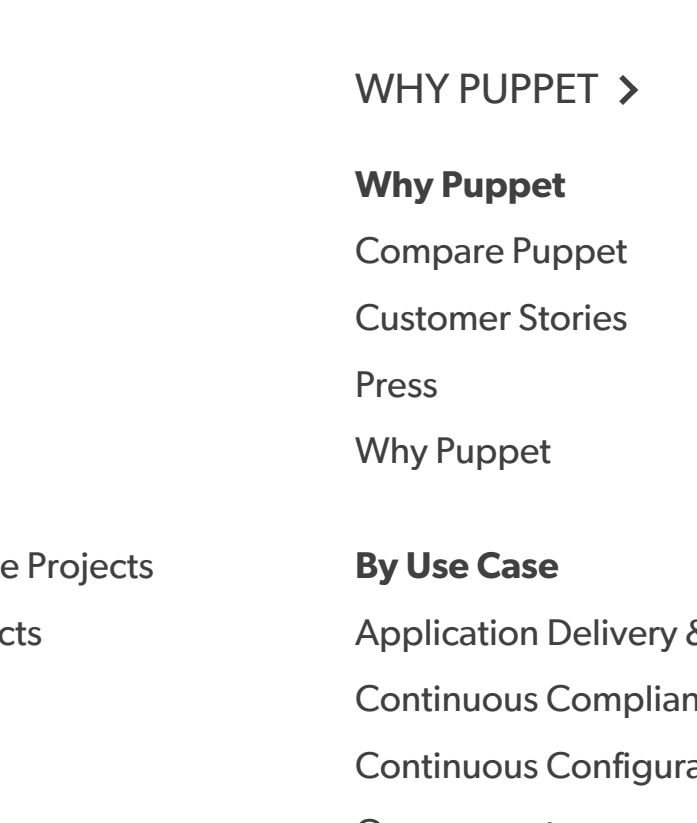
Principal Sales Engineer, Puppet by Perforce

Stephen is a Principal Sales Engineer at Puppet by Perforce. His years of experience in the Puppet ecosystem and decades in IT operations include roles as sysadmin, engineer, and architect for Unix, Linux, Virtualization, and Cloud technologies.

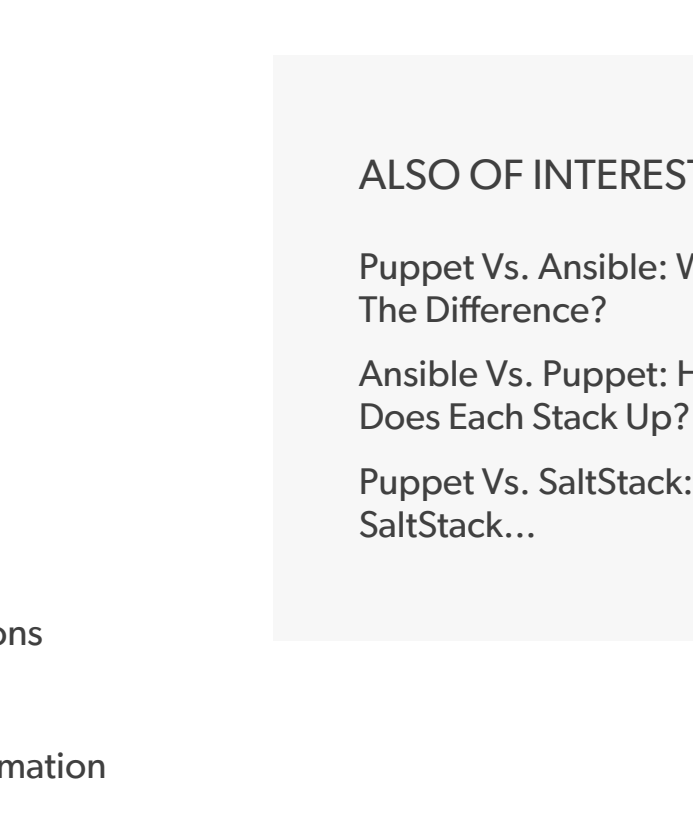
Recommended Blogs



DevOps in Government Environments



How to Learn Puppet the Easy Way — From...



Automation Technology in Government:...

PRODUCTS >

Puppet Enterprise

- [Overview](#)
- [Free Trial](#)
- [Pricing](#)
- [Product Demo](#)

Puppet Enterprise Extensions

- [Documentation](#)
- [Continuous Delivery For Puppet Enterprise](#)
- [Enterprise](#)
- [Content & Modules](#)

Product Resources

- [Documentation](#)
- [Integrations](#)
- [Knowledge Base](#)
- [Support](#)

Get Started

- [Request A Demo](#)
- [Try It Now](#)

COMMUNITY >

Puppet Forge

[Puppet Forge](#)

Open Source Projects

- [Open Source Puppet](#)
- [Bolt](#)
- [Contribute To Open Source Projects](#)
- [See All Open Source Projects](#)

Community

- [Community Calendar](#)
- [Community Overview](#)
- [Community Slack](#)
- [Puppet Champions](#)
- [Puppet Test Pilots](#)

Ecosystem

- [GitHub](#)
- [Integrations](#)
- [Puppet Developer Experience](#)
- [Trusted Contributors Program](#)

WHY PUPPET >

Why Puppet

- [Compare Puppet](#)
- [Customer Stories](#)
- [Press](#)
- [Why Puppet](#)

By Use Case

- [Application Delivery & Operations](#)
- [Continuous Compliance](#)
- [Continuous Configuration Automation](#)
- [Government](#)
- [IT Process Automation & Orchestration](#)
- [Patch Management](#)
- [Windows Infrastructure Automation](#)

SERVICES & TRAINING >

- [Professional Services](#)
- [Support](#)
- [Training & Education](#)
- [Courses](#)

RESOURCES >

- [Blog](#)
- [Customer Stories](#)
- [Events & Webinars](#)
- [On-Demand Webinars](#)
- [Papers & Videos](#)
- [Podcast](#)
- [Product Demos](#)

ALSO OF INTEREST

- [Puppet Vs. Ansible: What's The Difference?](#)
- [Ansible Vs. Puppet: How Does Each Stack Up?](#)
- [Puppet Vs. SaltStack: Best SaltStack...](#)

