Chef vs Puppet: A Detailed Comparison

Both Puppet and Chef are best-in-class configuration management tools that are leveraged to design, deploy, configure, and manage servers, infrastructure, etc. They are both, capable of automating multi-tier applications and competent enough to manage complicated apps.

What Is Configuration Management?

On a basic level, configuration management offers an abstraction layer between the infrastructure's basic formation and its chosen state by emphasizing important objectives rather than the monotonous responsibilities needed to attain them.

The major objective of configuration managers is to utilize comprehensive languages to understand and grasp all operations' necessities so that there is even knowledge sharing within different projects. For any newcomer in the team, it is ideal to have detailed know-how of the standardized tool rather than explaining to the team member about all the intricate details.

Benefits Of Configuration Management:

- Offers good control over the competence to review, approve, and make changes to a configuration item
- Automates identifies and guides configuration items through the project life cycle
- Ensures a lot of cost-effectiveness all through the project execution
- Provides a good quality output by measuring and tracking continuous integration components
- Addresses future needs in a better light and keeps the system up and running
- Increases reliability, stability, the effectiveness of the system

Puppet Configuration Management

"Unlike error-prone procedural scripts, which require you to define every step for configuring systems, Puppet Enterprise lets you simply define the desired state of your infrastructure and applications. Puppet continuously and automatically enforces that state, so you can spend more time innovating and less time-fighting fires." – Puppet

Puppet is used by thousands of companies worldwide some of them being Google, Red Hat, Siemens, Stanford, Harvard law school, and many more. It has a popular user-based contributing to its source code. It has been in vogue since 2005 and has come up with many versions with continuous improvement. It has been deployed in large infrastructures and possesses large user-driven documentation that suffices all queries. Puppet is based on Ruby and utilizes a tailor-made Domain Scripting Language nearer to JSON for working within it. It can execute on any platform that supports Ruby.

Core Components of Puppet:

- **Puppet Server** – the central server that accomplishes Puppet nodes
- **Puppet Enterprise Console** – a web GUI for report analysis and infrastructure resource control
- **Puppet Agent** – client software connected on managed nodes that permits harmonization with the Puppet Master
- **PuppetDB** – data storage service for the data formed

Pros And Cons of Puppet

Pros	**Cons**
• Well-built support system via Puppet Labs • Highly mature interfaces • Executes on all major OS • Simplistic installation and initial setup • Robust reporting competencies	• Knowledge of Ruby may be needed for progressive tasks • The model-driven approach may turn out to be tough to control • Tough for newer people to grasp

When To Use Puppet?

- Puppet is a preferred choice when the key concerns are stability and maturity. It is best for huge enterprises with a varied environment and different skills with the DevOps team
- Puppet, being the older one, offers more of traditional infrastructure support and hence can be opted by organizations who want to stick to the original traditional outlook
- Puppet is good if your team has a range of skills on the DevOps team

Chef Configuration Management

"Chef is the leader in Continuous Automation software, an innovator in application automation and one of the founders of the DevOps movement. Chef works with more than a thousand of the most innovative companies around the world to deliver their vision of digital transformation, providing the practices and platform to deliver software at speed." – Chef

Chef is an automation tool that offers a mechanism to define infrastructure as a code. Instead of using manual procedures, it believes in managing infrastructure by writing code. It utilizes the Ruby language for writing the configuration items. It automates infrastructure configuration, app deployment, and configurations managed across the network.

Core Components of Chef:

- **Chef Server** – The main hub where Chef propagates and stores recipes and cookbooks
- **Chef Client** – Performs configuration tasks on the local machine and is installed on every node being managed
- **Workstation** – Permits selected workstations to author/test/maintain cookbooks and upload them to Chef Server
- **Chef Analytics** – A stage that delivers actions and runs history, real-time reporting, and notifications around Chef mechanization activities
- **Chef Supermarket** – An open-source directory of community-contributed cookbooks

Pros And Cons of Chef

Pros	Cons
• An enriched assortment of modules and configuration controls • Highly flexible and skilful because of code-driven approach • Strong version control competences	• Sharp learning curve if Ruby isn't known • Not highly simplistic due to the big codebase and complex environment • Doesn't support push functionality

When To Use Chef?

- Chef is a preferred choice when you have skilled resources in Git and Ruby. It is best for teams that are focused on development and for enterprises that are looking for a modernized environment

- You can choose Chef if you are known to Git and Ruby.
- Chef, being the younger one, showcases more modernization and hence can be opted by organizations who are looking for an advanced feel

Chef vs Puppet – The Similarities

Since both look at the same fundamental objective of configuration management, there are certain attributes that are found to be existing in both, here are they:

- Both help development and operations teams manage applications and infrastructure
- Both are highly mature technologies that are extensively installed, possess detailed documentation
- Both have dynamic user groups and strong SLAs (service level agreements)
- Both have respectable and big brands as their clients. Puppet is used by Red Hat, Siemens, Salesforce, Sony, Google, etc. Chef is used by Facebook, Mozilla, Expedia, Rackspace, Xero, Disney, GE, etc.
- Both generally work in client-server modes
- Both are highly scalable and have a high interoperability
- Setup and management aren't very easy in either
- Both have a backup server to act immediately if the main server fails

Chef vs Puppet – The Differences

Puppet	**Chef**
Accommodates careful system administrators	Caters to imaginative developers
Focuses on users not making mistakes and is more protective	Focuses on users developing new things and helps take risks
Uses a declarative language like JSON or XML	Uses an imperative language as if its wholly featured Ruby
The user creates manifests and modules	The user creates recipes and cookbooks
Puppet uses standardized tools such as RSpec and Cucumber for testing	Chef uses dedicated solutions such as Test Kitchen for testing
The module directory that offers support is PuppetForge	The module directory that offers support is Supermarket

Puppet Masterworks only on Linux/Unix, but Puppet Agent also works on Windows	Chef Server works only on Linux/Unix, but Chef Client and Workstation can be on Windows also
Puppet is a model-driven architecture	Chef is based on a procedural architecture
Since Puppet is an older technology than Chef, its traditional infrastructure operations are well-liked by the certain user community	Since Chef is the younger one, it offers a modern approach to configuration management which is appreciated by a specific set of users
Puppet leverages its own non-standard custom programming language with limiting rules and hence isn't that extensible	Chef uses the DSL but when that isn't enough, it leverages the potential of Ruby to the maximum

Puppet vs Chef: The Wrap-up

As we read through the details of Chef vs Puppet, we understand the importance of both. Both are competitive, good, popular and are in line with the latest DevOps-related needs. Both are upgrading themselves to fit in the latest groove. Both have matured immensely and hence which one to choose is entirely up to the organization's basic needs and workflows.

Source: https://www.spec-india.com/blog/chef-vs-puppet-a-detailed-comparison