



Chef is the infrastructure automation engine for modern software-driven organizations. Ship your ideas faster with Chef.

Today, every company is a software company. And software changes constantly. Don't let cumbersome, manual infrastructure change management processes slow your company down. By automating your infrastructure changes, you can achieve consistent, repeatable, and fast software delivery to any data center or cloud environment.

Using Chef, you can:

- **Replace manual processes, runbooks, or brittle scripts with easy-to-understand automation code.** The Chef language is designed to be easy to read and write, and helps you declare & enforce a system's intended state, instead of having to memorize arcane commands

BEFORE CHEF	AFTER CHEF
<pre>yum -y install httpd /sbin/chkconfig httpd on /sbin/service httpd start</pre>	<pre>package 'httpd' do action :install end service 'httpd' do action [:enable, :start] end</pre>

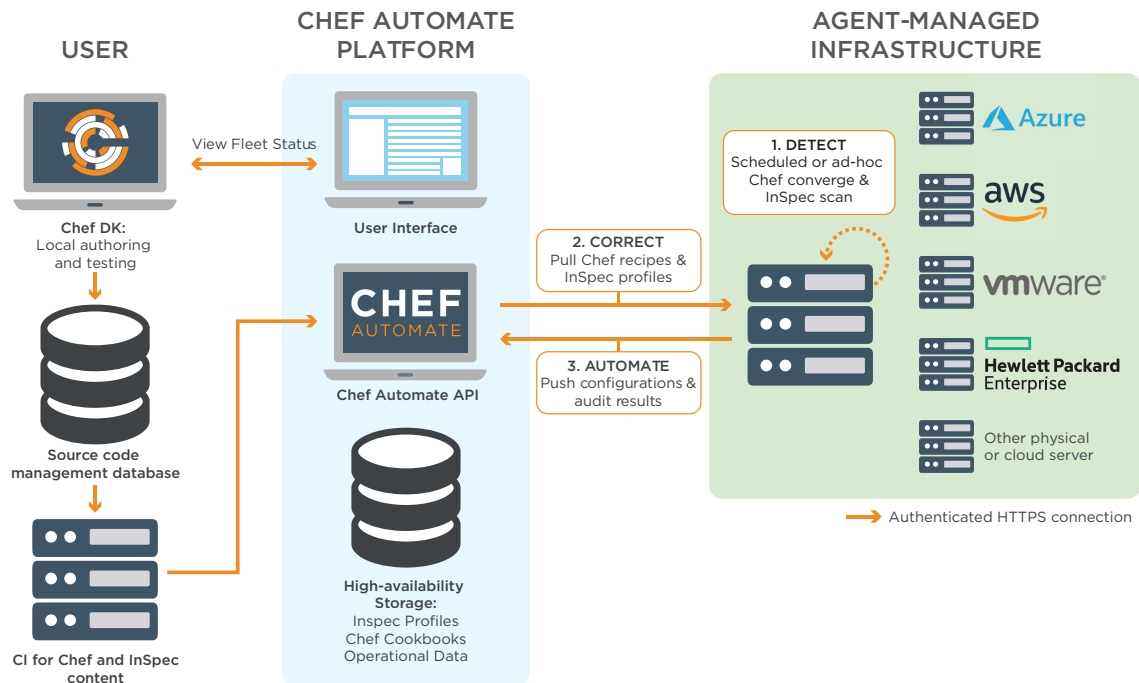
and options. The same language is used across Windows, Linux, UNIX, making it easy to understand how to perform similar actions across vastly different operating systems.

- **Reduce risk by validating infrastructure changes before pushing them to production.** Because the Chef language is engineered on top of a full programming language, it is infinitely extensible to your environment, and the code you write can be tested for both correctness and quality before deployment. Use Chef's rich ecosystem of tools like Test Kitchen, ChefSpec and Foodcritic to quickly simulate real environments and see the impact your changes make before they impact production.
- **Adopt the cloud quickly by accelerating the setup and maintenance of ephemeral servers.** Achieve the velocity that new business initiatives demand. Couple the cloud's on-demand infrastructure with last-mile configuration using Chef to ensure new systems are set up properly the first time, regardless of whether they number in the dozens or thousands.



How It Works

Chef operates in a client-server architecture. An agent, the Chef Client, runs on each managed node, periodically connecting to a Chef Server to download and evaluate configuration code, known as recipes. If no changes are necessary, Chef does not modify the system.



Systems administrators or developers write recipe code on their workstations and submit it through a continuous integration/continuous delivery (CI/CD) pipeline, just like application software, to ensure that only high-quality code makes it to the Chef Server. InSpec controls can be used as part of the process to evaluate environments for correctness and compliance.

Chef and Chef Automate

Chef Automate creates real-time operational dashboards providing insight into the events generated by each Chef Client run. With Chef Automate's dashboards, you can:

- Visualize and monitor fleet-wide health across hundreds, thousands, or even tens of thousands of nodes, and filter results by physical or logical environment.
- See the successes and failures of your Chef Client runs, drill into their results in aggregate or individually, and diagnose problems quickly.
- Get alerted to errors immediately using Chef Automate's chat and webhook notifications system.

Finally, if you are using InSpec, you can display your configuration management data alongside compliance data to see how infrastructures changes impact compliance and vice-versa.

Try Chef Today

Ready to see how Chef can accelerate your company's software time-to-market? Try it today at learn.chef.io.

