

Chef Workstation

Create, Test and Deliver
Automation to Any System

PRODUCT GUIDE



Table of Content

1. Overview / 3

Chef Policy-Based Automation Architecture / 3

Chef Workstation vs. ChefDK / 5

2. Chef Workstation Core Components / 6

Chef Tools: Chef Infra, Chef InSpec and Chef Habitat / 7

Chef Language, Resources, Helpers and Policyfiles / 8

Testing and Upgrade Tools / 9

3. What's New in Chef Workstation / 10

Improved target platform OS support / 12

4. Chef Workstation Top Use Cases / 12

Automate Configuration Management / 12

5. Getting Started with Chef Workstation / 15

Download Chef Workstation / 15

Installation / 16

Setting up Workstation / 16

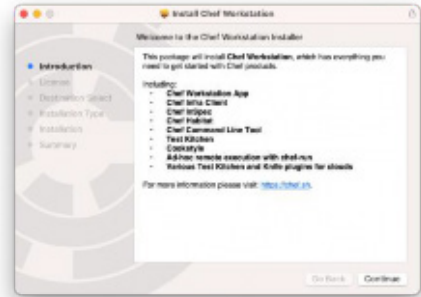
6. Additional Resources / 19

Overview

Chef Workstation gives users all the tools they need to get started with Chef all in one easy-to-install package. Users can execute ad-hoc remote configuration tasks, perform remote scanning, create cookbooks and compliance profiles, test software and dependencies and much more. Chef Workstation provides a local developer experience and can be run on Linux, macOS, or Windows systems.

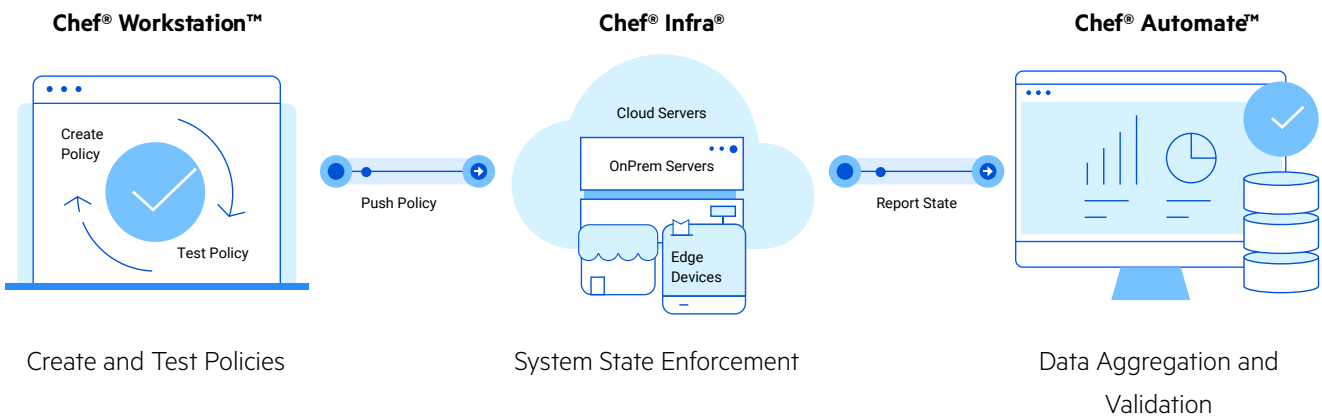
Chef Workstation includes:

- **Chef Tools:** Chef Infra, Chef InSpec and Chef Habitat
- **Chef Language:** Pre-built resources for managing systems as well as helpers to make authoring and distributing cookbooks easy
- **Embedded applications:** Ruby comes with Workstation, so there's no need to install anything else
- **Testing and Upgrade Tools:** Test Kitchen, Chef Cookstyle and Chef Upgrade Lab



Chef Policy-Based Automation Architecture

Using Chef to automate configuration management allows DevOps teams to define policies that are repeatable, consistent and reusable. The result is increased business agility and security because all systems and resources are continuously and automatically evaluated, corrected, and modified.



Chef Workstation	Chef Infra Server and Client	Chef Automate
<p>Reduce risks by iterating on policy changes before pushing them to production.</p> <ul style="list-style-type: none"> Chef Tools: Chef Infra Client, Chef InSpec, Chef Habitat, Chef Cookstyle, Test Kitchen, and knife Chef Language: Pre-built resources for managing and securing systems as well as helpers that make authoring and distributing policy easy 	<p>Enforce policy by converging the system to the state declared by the various resources.</p> <ul style="list-style-type: none"> Planned, unstructured and policy-based updates Compliance Phase, which brings together Infra and InSpec Dynamic behavior support Ephemeral resource management Data collection 	<p>View and validate intended and actual state across all systems.</p> <ul style="list-style-type: none"> Real-time interactive dashboards Role-based access controls Third-party integrations Data APIs Chef Infra Server management

Over the past 10 years, a lot has changed within the Chef Infra product portfolio. We've continued to evolve our solutions to make them easier to use, adding new functionality and increasing scalability. The table below provides a summary of key advancements made within the portfolio in the last 5 years.

Chef Past		Chef Present
Ruby	Language	Chef Language and Pre-Built Resources Helpers
ChefDK, ChefSpec, and Foodcritic	Developer Kit	Chef Workstation, Test Kitchen, and Chef Cookstyle
Roles, Environments, and Audit Cookbook	Content and Compliance	Policyfiles, Policy Groups and Chef Client Compliance Phase
Stand Alone Chef Server and Chef Manage	Management	Chef Automate with Integrated Chef Server
Adhoc Testing	Testing	Test Kitchen and Test Driven Development Adoption
Manually Pushing Changes	Deployment	CI/CD Automated Pipelines, GitHub Actions, and Pull Requests

Table 1: Chef Past and Present: Summary of Advancements Made in the Last 5 Years

Chef Workstation vs. ChefDK

ChefDK End of Life

ChefDK was first shipped in 2014 and on **Dec 31, 2020 reached end-of-life status**. Chef Workstation launched in 2018 and has replaced ChefDK. Chef Workstation includes all the tools you're familiar with in ChefDK, plus much more. We've packaged new tools for working with VMware, executing ad-hoc jobs, updating your cookbooks and policyfiles, and bootstrapping systems. For more information read the blog "Goodbye ChefDK, Hello Chef Workstation".

Chef Workstation Key Features and Enhancements vs. ChefDK

Added Tools and Support	Updated Components	Updated Cloud SDKs
<ul style="list-style-type: none">• Include_profile, include_waiver, include_input and other resources for integrating Infra and InSpec• Knife support for VMware vCenter• Test Kitchen support for VMware vCenter, VMware vRA, and OpenStack• Chef run for adhoc Chef Infra execution• Support for generating InSpec profiles, waivers, and inputs with Chef generate• Support for simplified Policyfile format to easier policy creation and promotion• Packages for Windows 11, macOS 12, and macOS 11/12 on M1 architecture• Improved secrets management with Vault	<ul style="list-style-type: none">• Chef Infra Client 18 vs. 15 Chef CLI 5.6• Cookstyle 7.32 vs. 5.23• InSpec 4.52 vs. 4.24• Habitat version 1.6• Test Kitchen 3.2 vs 2.8.0• Updated Test Kitchen drivers for Azure, Vagrant, EC2, Google Cloud, Docker, Hyper-V, and DigitalOcean	<ul style="list-style-type: none">• GCE• DigitalOcean• AWS• Azure

The following tools have been removed from ChefDK/ Workstation and are no longer being actively developed by Chef:

- foodcritic (replaced with Chef Cookstyle)
- chef-api
- chef-sugar
- cucumber
- knife-spork
- push jobs client
- knife-push
- stove

ChefDK reached end of life at the end of 2020 and is no longer being maintained. CVEs with known issues that impact ChefDK include:

- OpenSSL 1.0.2x
- Ruby 2.6.6
- Git 2.26.2
- curl 7.73
- libarchive 3.4.3
- Nokogiri 1.10.1

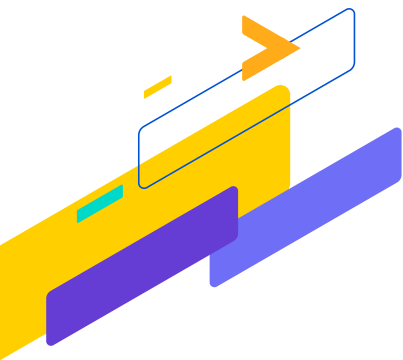
Upgrading to Workstation

To upgrade from ChefDK to Chef Workstation, all you need to do is install the Chef Workstation package available at downloads.chef.io. The Chef Workstation package will replace your existing ChefDK installation, and you'll be ready to enjoy all the new functionality provided in Chef Workstation.

Chef Workstation Core Components

Chef Workstation includes every tool you need for infrastructure, compliance, and application delivery automation within your organization. Everything you need to get started — ad hoc remote execution, remote scanning, configuration tasks, cookbook creation tools as well as robust dependency and testing software — all in one easy-to-install package. The Workstation toolkit includes:

- Chef Tools: Chef Infra, Chef InSpec and Chef Habitat
- Chef Language, Resources, Helpers and Policyfiles
- Testing and Upgrade Tools



Chef Tools: Chef Infra, Chef InSpec and Chef Habitat

Chef Infra

[Chef Infra](#) is a configuration management tool for defining [infrastructure as code \(IAC\)](#), making the process of managing configurations and system state automated and testable. Chef Infra takes a policy-based approach that builds upon the principles of test-driven development and idempotency. System configuration and application change testing can be done in parallel, so system states are easily enforced and monitored across all infrastructure and teams. With Chef Infra, users define configurations once and apply them across mixed fleets of Linux, macOS, and Windows systems, regardless of OS version and architecture.

Chef Workstation is used to create, test and manage Chef Infra cookbooks, InSpec profiles and Habitat plans. It includes Test Kitchen for code testing using Docker, AWS, Azure, Vagrant and other providers, and Cookstyle, a powerful linting tool with autocorrect capabilities.

A Chef Infra Cookbook is the fundamental unit of configuration and policy automation distribution. Cookbooks are used to describe the system resources under management, such as files, templates, ports, and software packages. Resources are defined in Chef Infra recipes that describe in human-readable code the desired state of each system. The code describes the desired state and then the Chef Infra Client automatically configures the system.

Learn more about Chef Infra features and benefits by downloading the [Chef Infra Client 17 Product Guide](#).

Chef InSpec

Chef InSpec is a compliance management tool for defining [compliance as code \(CaC\)](#), making the process of managing compliance and system secure state automated and testable. Chef InSpec makes it easier than ever to maintain and enforce compliance across the enterprise, with standards-based audit and remediation content, easily tuned baselines to adapt to the organization's needs, and visibility and control across hybrid and multi-cloud environments.

Using Chef Workstation, Chef InSpec users create simple controls to verify the expected state of systems against their current state. Users can create a wide variety of profiles to validate Chef cookbook logic, verifying both hardware and software resources, or take advantage of publicly available profiles and Chef premium content. With profiles now part of cookbooks, it's easier than ever to apply them as part of your DevOps Policy as code workflows, from iteration testing with Test Kitchen to full enterprise deployments. InSpec profiles are the ideal way for your DevOps teams to confidently write, test and deploy Chef code.

[Learn more about Chef InSpec](#)

Chef Habitat


Design and package applications as code and deliver them to any environment with Chef Habitat. Habitat enables DevOps teams to build continuous delivery pipelines across all applications by automating the process of defining and packaging applications through 'artifacts' that can be seamlessly deployed anywhere – VMs, physical servers or in the cloud. Habitat ensures your agile delivery practices are free of obstacles and fast, by storing automation as coded artifacts that are easy to search, share, and deploy. Habitat enables fully automated release processes for faster deployments with the flexibility to rollback releases as needed. Efficiency is maximized through continuous automation of processes that define, package and deploy applications components – all through unified platform agnostic code.

[Learn more about Chef Habitat](#)

Chef Language, Resources, Helpers and Policyfiles

The [Chef Infra Language](#) is a comprehensive systems-configuration language with resources and helpers for configuring operating systems. The Chef Infra Language is based on Ruby, allowing you to utilize the power of Ruby when the built-in language doesn't meet your needs out of the box. With the Chef Language, users define configurations once and apply them across mixed fleets of Linux, macOS and Windows systems, regardless of OS version and architecture.

- **Resources:** Resources are used for configuring components such as packages, files, directories, ports, or firewalls. Today, Chef Infra Client ships with about 170 resources for common automation tasks such as user, file, kernel_module and windows_task.

- 
- **Helpers:** Helpers enable users to make configuration decisions based on operating systems, clouds, virtualization hypervisors, and more.
 - **Policyfiles:** Policyfiles are the best way to handle dependencies and change management across your Chef Infra managed infrastructure. They combine the very best parts of Roles, Environments, and Berkshelf into a single workflow. A single chef install command bundles multiple cookbooks, profiles and dependencies into an immutable object. Immutable Chef Infra Policyfiles cannot be changed once bundled, ensuring your configurations change only when you want them to. The Chef Infra Client no longer recalculates dependencies at the start of every run, making them faster and more efficient. For more information view the [Policyfile documentation](#)

Testing and Upgrade Tools

Chef Workstation comes with linting and testing tools to help you create clean code. If you're upgrading from an earlier version of the Chef Infra Client, the Chef Upgrade Lab helps you take the necessary steps to update your older code.

- **Cookstyle:** Cookstyle is a code analysis tool built upon [RuboCop](#) and ships as part of Chef Workstation. It replaced Foodcritic in September 2019. Today, Chef Cookstyle includes more than 250+ Chef Infra specific rules (called cops) that catch common cookbook coding mistakes, clean up portions of code that are no longer necessary, and detect deprecations that prevent cookbooks from running on the latest releases of Chef Infra Client.
- **Test Kitchen:** Test Kitchen is an independent open-source project sponsored by Chef that provides real-world test environments to execute infrastructure code on one or more platforms in isolation. Test Kitchen is Chef's integration testing tool of choice for cookbooks. It's included as part of Chef Workstation and used by all Chef-managed community cookbooks.
- **Chef Upgrade Labs:** Chef Upgrade Lab is a process guide for upgrading nodes from legacy Chef Infra Client 12 through 17 to current Chef Infra Client 18 using a powerful set of tools. The [Upgrade Lab Guide](#) shows users how to query their existing infrastructure to identify a single node to upgrade, capture the state of that node to a local development workstation, iterate locally on cookbook changes to support the latest version of Chef Infra Client, deploy the updated cookbook to a new Chef Infra Server instance, and finally upgrade the installed version of Chef Infra Client on the node.

What's New in Chef Workstation

Chef Workstation has released some major enhancements that include:

- Compliance Phase tooling to integrate Chef InSpec and Chef Infra
- Ruby 3
- Policyfile improvements
- 29 new Cookstyle cops
- Test Kitchen platform driver improvements
- Improved target platform OS support

Compliance Phase tooling

Updated Chef generators create compliance/ subdirectories for inputs, profiles and waivers.

Several new resources have been added and existing resources have been enhanced.

```
my_cookbook/  
  compliance/  
    inputs/  
      my_inputs.yml  
    profiles/  
      my_profiles/  
        controls/  
          inspec.yml  
      waivers/  
        control_waiver.yml  
  metadata.rb
```

Ruby 3

Chef Workstation now takes advantage of the latest version of Ruby as the core of the Chef Language.

Policyfile Improvements

Simplified Policyfile.rb config format with new metadata configuration option. Also, chef export allows defining a Policy Group in the export with `-policy_group`.

29 new Cookstyle Cops

We continue to enhance Cookstyle's linting capabilities, and we've added new cops to enhance existing capabilities and introduce new ones.

Test Kitchen Improvements

Improvements to Test Kitchen build on massive updates made in the release of Chef Infra Client 17 and make it easier than ever to test cookbooks and policies using Docker, Vagrant or cloud environments. Users can spin up multiple environments and OSes at once and add suites to include cookbooks and policies to run automatically. We've created Docker (Dokken) images that simulate true OS environments so you can test with confidence.

```
1 ---
2 driver:
3   name: dokken
4   use_sudo: false
5   privileged: true # allows system services to start
6
7 provisioner:
8   name: dokken
9
10 transport:
11   name: dokken
12
13 verifier:
14   name: inspec
15
16 platforms:
17 - name: ubuntu-22.04
18   driver:
19     image: dokken/ubuntu-22.04
20     pid_one_command: /bin/systemd
21 - name: centos-8
22   driver:
23     image: dokken/centos-8
24
25 suites:
26 - name: default
27   verifier:
28     inspec_tests:
29     - compliance/policies/myproject
30
```

To learn more about testing with Docker read the blog and watch the on-demand webinar [Chef Infra Best Practices: #3 Testing Chef Infra Cookbooks Fast with Docker](#).

Improved target platform OS support

- Ubuntu 20.04
- Windows 11
- Windows 2022
- macOS 12
- Debian 11
- macOS
- arm packages

Chef Workstation Top Use Cases

Chef Workstation is the developer kit for working with all aspects of Chef and Chef code. It includes tools for local testing, linting, and interacting with Chef Infra Server. Chef Workstation allows organizations to deploy a secure, compliant, and scalable infrastructure across cloud, VMs, or physical environments. Here are some of the use cases of Chef Workstation:

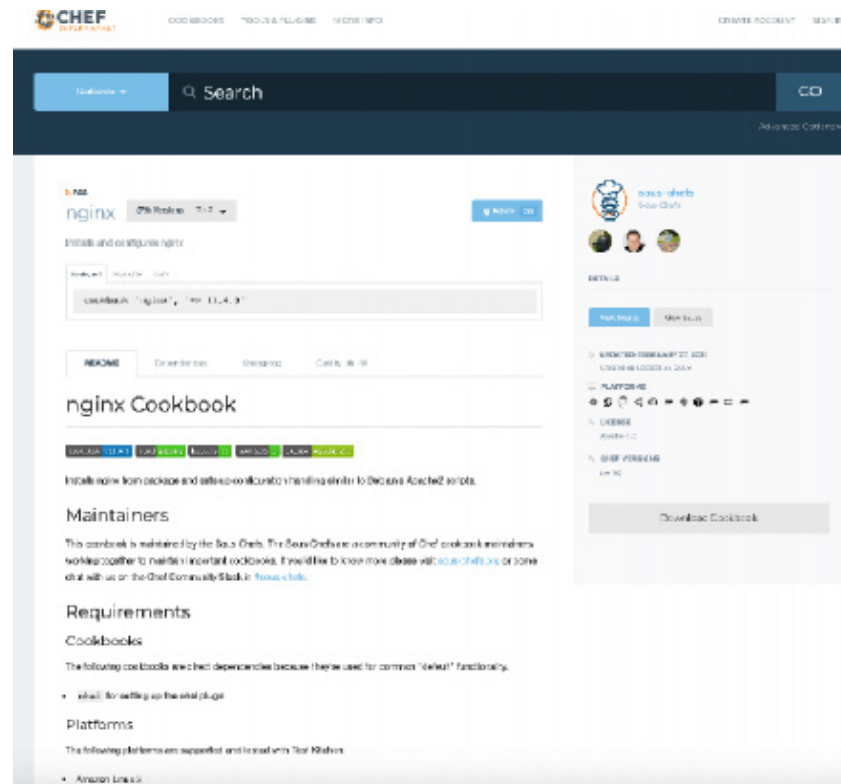
Automate Configuration Management

Chef Infra within the Workstation console helps automate infrastructure configuration management across large, distributed networks of IT servers in physical, virtual or cloud environments. Our Policy as Code approach allows configuration policies to be defined in the form of recipes and cookbooks at a centralized location. Defined once, these configuration policies are then applied to all nodes in the network through Chef Clients that reside on these nodes. Platform agnostic, highly flexible policies that allows users to modify configuration changes as desired and apply it safely and consistently across all endpoints. Additionally, quickly configure servers directly over SSH or WinRM and execute ad-hoc tasks on multiple hosts in parallel to quickly configure groups of servers.

```
chef@workstation:nginx$ chef-run ssh://root@ubuntu01 recipes/default.rb -l ~/.ssh/id_rsa
[✓] Packaging cookbook... done!
[✓] Generating local policyfile... exporting... done!
[-] Applying nginx::default from /Users/tonello/chef-repo/cookbooks/nginx to target.
[-] [ubuntu01] Applying nginx::default...
```

Get a Head Start on Chef Cookbooks and Profiles with Supermarket

Visit [Chef Supermarket](#) to find a library of partner and community-maintained configuration policy or use Chef's declarative resources to create custom resources for your organization. Choose from a variety of industry-benchmarked content for infrastructure management offered by Chef or create your own custom policies.



Test-Driven Development

Chef Workstation includes Cookstyle and Test Kitchen – tools that allow automated testing of your configuration and compliance policies. Users can simulate a real environment by provisioning live systems to test and verify their code. Chef InSpec testing enables provisioned systems (containers, VMs, cloud servers and edge devices) to be configured and tested. Policies and configurations that pass these tests can be confidently applied to production systems.

```
chef@workstation:nginx$ kitchen test
----> Starting Test Kitchen (v3.2.2)
----> Creating <default-ubuntu-2004>...
  Creating kitchen sandbox at /Users/tonello/.dokken/kitchen_sandbox/507eab7ab3-default-ubuntu-2004
  Creating verifier sandbox at /Users/tonello/.dokken/verifier_sandbox/507eab7ab3-default-ubuntu-2004
  Building work image..
  Creating container 507eab7ab3-default-ubuntu-2004
  Finished creating <default-ubuntu-2004> (0m3.02s).
----> Converging <default-ubuntu-2004>...
  Creating kitchen sandbox in /Users/tonello/.dokken/kitchen_sandbox/507eab7ab3-default-ubuntu-2004
  Preparing dna.json
  Resolving cookbook dependencies with Berkshelf 7.2.2...
  Removing non-cookbook files before transfer
  Finished creating <default-ubuntu-2004> (0m3.02s).
----> Converging <default-ubuntu-2004>...
  Creating kitchen sandbox in /Users/tonello/.dokken/kitchen_sandbox/507eab7ab3-default-ubuntu-2004
  Preparing dna.json
  Resolving cookbook dependencies with Berkshelf 7.2.2...
  Removing non-cookbook files before transfer
```

[Learn more](#)

Automate Compliance and Security Audits, and Remediation of Vulnerabilities

Workstation bundles Chef's popular Security and Compliance Automation tool, InSpec, to define security and compliance policies, and automatically run continuous audits to detect and remediate vulnerabilities. Platform-agnostic code allows compliance checks on infrastructure configurations, security policies and provisioned resources. Chef InSpec provides a local agent for host-based assessments, and full remote testing support via SSH and WinRM. Chef Automate also includes industry-benchmarked premium content to help you and your teams get started while providing the flexibility to modify policies to suit your needs.

Example:

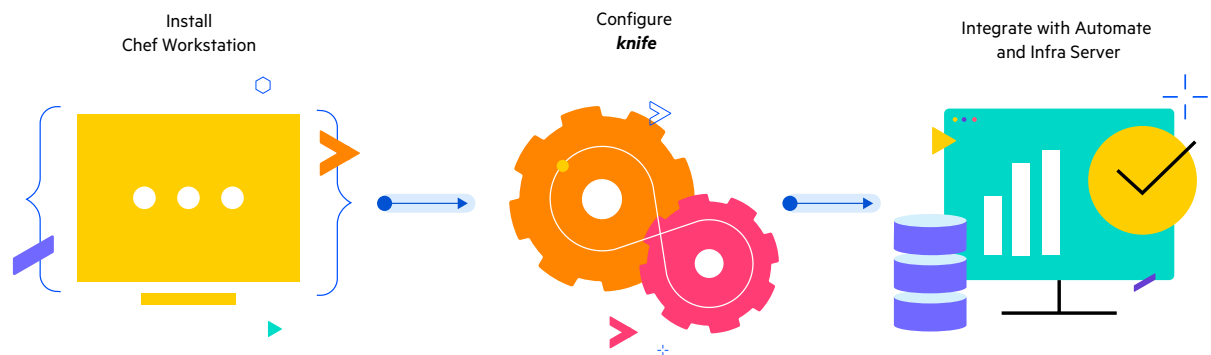
- Scan servers to verify the NGINX is installed and running
- Scan cloud resources for compliance status
- Remediate scan failure by Installing and configuring NGINX

```
chef@workstation:nginx$ inspec exec compliance/profiles/nginx/ -t ssh://root@ubuntu01 -l ~/.ssh/ld_rsa
Profile: Nginx profile (nginx)
Version: 0.1.2
Target:  ssh://root@ubuntu01

* nginx-config: Nginx configuration (4 failed)
  ✓ Port 80 is expected not to be listening
  * Port 443 is expected to be listening
    expected 'Port 443.listening?' to be truthy, got false
  * System Package nginx is expected to be installed
    expected that 'System Package nginx' is installed
  * HTTP GET on https://localhost:443/ status
```

Getting Started with Chef Workstation

Workstation is a one-stop solution for developers to get their infrastructure automation processes under control. Here's a quick-start guide that explains how to install, configure, get started with the Workstation desktop application for ad hoc remote execution, remote scanning, configuring tasks, cookbook creation, dependency mapping and testing policies.



Download Chef Workstation

Chef Workstation is available for download on the following operating systems:

- Amazon Linux
- Apple macOS
- Debian
- Microsoft Windows
- Red Hat Enterprise Linux / CentOS / AlmaLinux / Rocky Linux
- Ubuntu

[Read More](#)

System Requirements for Workstation include:

- 4 GB RAM
- 8 GB HDD
- Some additional memory and storage to create virtual test environments with Test Kitchen

[Read more](#)

The Workstation developer kit includes the following tools bundled within it:

- Chef Infra Client
- Chef InSpec
- Chef Habitat
- Chef Command Line Tool
- Chef Cookstyle
- Test Kitchen
- Other tools to help in policy authoring

[Download Chef Workstation](#)

Installation

1. The Chef Workstation installer must run as a privileged user.
2. Select and download the package appropriate for your distribution and follow the instructions on screen to accept license and install Workstation.
3. On macOS, Chef installs to `/opt/chef-workstation/`. Alternatively, you can install using the [Homebrew](#) package manager: `brew install --cask chef-workstation`
4. On Windows, Chef installs to `C:\opscode\chef-workstation\`. Alternatively, you can install using the [Chocolatey](#) package manager: `choco install chef-workstation`
5. On Linux, use the links from the downloads page or use the package manager:
 - Red Hat Enterprise Linux: `yum localinstall chef-workstation-21.2.278-1.el8.x86_64.rpm`
 - Debian/ Ubuntu: `dpkg -i chef-workstation_21.2.278-1_amd64.deb`
6. To avoid interference with other applications running on the target machines, we do not recommend changing the default installation locations.
7. To verify your installation, run: `chef -v`. This will return versions of all installed Chef tools.

[Read more about installing/ uninstalling Chef Workstation](#)

Setting up Workstation

1. Prerequisites

- [Install](#) Chef Workstation
- A running instance of [Chef Infra Server](#) or [Hosted Chef Server](#)
- `client.pem` file, if Chef Manage or Hosted Chef is not in use

2. Configure Ruby environment

If you want to use Ruby as your primary language for Chef Policies, we recommend using the version installed with Workstation. If you use Ruby for software development outside of Workstation, you can continue as usual. On Windows, Chef Workstation includes a desktop shortcut to a PowerShell prompt already configured for use.

[For macOS and Linux, click here for step-by-step instructions](#)

3. Set up chef-repo for cookbook storage

When setting up Chef Workstation for the first time, we recommend a Chef Infra repository for saving everything you need to define your infrastructure. The repository should be synchronized with git or any other version control system. This repository will include:

- Cookbooks (recipes, attributes, compliance profiles, custom resources, libraries, files, and templates)
- Data bags
- Policyfiles

- Create your own Chef Infra repo using the `chef generate repo <repo-name>` command.
- Use the `knife` and `chef` commands to upload data to Chef Infra Server from your local repo.
- Chef Infra Client will use this uploaded data to manage all nodes registered with the Chef Infra Server.

[Read more](#)

4. Chef credential setup

The first time you run Chef Workstation, a `.chef` directory is created. This is where all Workstation configuration and client keys are stored.

Note: Setting up Chef Workstation as a Chef Infra Server Administrator

You will need to separately manage users with the [Chef Infra Server CLI](#) or the Manage UI. A user-specific RSA client key is generated every time a new user is created and this Infra Server credential. should be shared securely with the Workstation user.

Setting up Chef Workstation as a Chef User

You will need the Chef Infra Server user key that your server administrator creates for you and it should be placed in your \$HOME/chef directory. This key is an RSA private key in the .pem format.

For more details on creating credential files and setting up private keys, try the links below.

[How to create your user credentials file](#)

[How to setup your client private key](#)

5. Verification of Client-Server communication

Run the following command:

```
knife configure init-config
```

```
knife ssl fetch
```

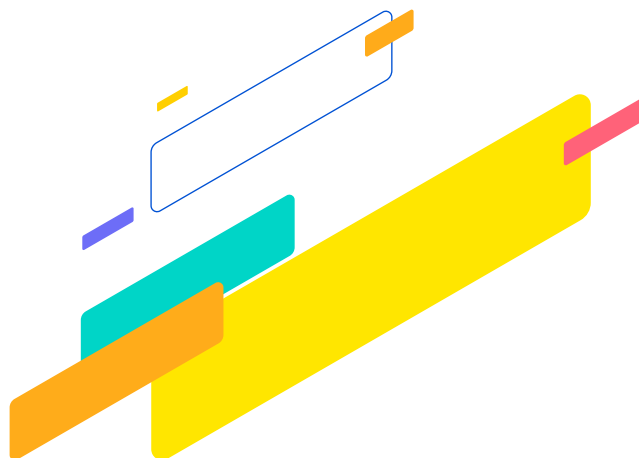
```
knife ssl check
```

```
knife client list
```

These steps will set up secure communication between your Workstation and the Chef Infra Server. The configuration command creates a \$HOME/chef/credentials file that contains information about your Infra Server, your user and other configurations. When the configuration is complete, run the other knife commands to verify the secure connection and return a list of connected clients which will look something like this:

```
chef_machine
```

```
registered_node
```



Additional Resources

Configuring Chef Workstation

Chef Workstation's configuration is managed in `$HOME/chef-workstation/config.toml`. Chef Workstation will create `config.toml` the first time you use `chef-run`, if one does not already exist.

[Chef Workstation Configuration Guide](#)

Troubleshooting Chef Workstation

The link below will help understand how to troubleshoot commonly known issues while setting up Chef Workstation.

[Chef Workstation Troubleshooting Guide](#)

About Progress

Dedicated to propelling business forward in a technology-driven world, Progress (NASDAQ: PRGS) helps businesses drive faster cycles of innovation, fuel momentum and accelerate their path to success. As the trusted provider of the best products to develop, deploy and manage high-impact applications, Progress enables customers to build the applications and experiences they need, deploy where and how they want and manage it all safely and securely. Hundreds of thousands of enterprises, including 1,700 software companies and 3.5 million developers, depend on Progress to achieve their goals—with confidence. Learn more at www.progress.com

2022 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
Rev 2022/06 RITM0163080

Worldwide Headquarters

Progress Software Corporation
15 Wayside Rd, Suite 400, Burlington, MA01803, USA
Tel: +1-800-477-6473

 facebook.com/getchefdotcom
 twitter.com/chef
 youtube.com/getchef
 linkedin.com/company/chef-software
 learn.chef.io
 github.com/chef
 twitch.tv/chefsoftware