

# Getting Started with Chef Automate

Install and Configure Chef Automate and Infra Server

## Contents

With the launch of Chef Infra Client 17 in April, there's never been a better time to dig into the Chef ecosystem, infrastructure automation, continuous delivery and configuration management with Chef Automate, the dashboard tool that allows you to see all your Chef activity in real-time.

<b>Overview</b> .....	<b>2</b>
<b>Deploy in AWS and Azure Marketplaces</b> .....	<b>3</b>
<b>Deploy Chef Automate and Chef Infra Server Manually</b> .....	<b>3</b>
1. <b>Install, Configure and Run Chef Automate</b> .....	<b>4</b>
2. <b>Deploy Chef Automate with chef-automate</b> .....	<b>4</b>
3. <b>Log into the Chef Automate Dashboard</b> .....	<b>5</b>
4. <b>Add your Chef Infra Server to the Chef Automate Dashboard</b> .....	<b>5</b>
5. <b>Configure Your Workstation to Communicate with Your New Server</b> .....	<b>6</b>
<b>Next Steps</b> .....	<b>7</b>

## Overview

[Chef Automate](#) is the dashboard, data aggregation, and analytics layer for all Chef products, including [Chef Infrastructure Management](#), [Chef Compliance](#), [Chef App Delivery](#), [Chef Desktop](#), and [Chef Enterprise Automation Stack](#). It allows developers, operators, and security engineers to collaborate effortlessly to deliver application and infrastructure changes at speed and scale. Chef Automate provides actionable insights across multiple data centers and cloud providers where your nodes live.

In this Chef Guide, I'll explain how to get up and running with Chef Automate in the public cloud using AWS or Azure, or in an-prem lab. You'll get a taste for how it provides insight into Chef Infra, Chef InSpec and Chef Habitat. If you have a license for any Chef product, you can use that license to activate your installation. If not, you can run a fully functional Chef Automate stack free for 60 days.

If you're using the AWS or Azure marketplaces, you can launch a fully operational Chef Automate/Chef Infrastructure Server in minutes with one click.

## Deploy in AWS and Azure Marketplaces

Please note that the marketplace images are meant for demo and testing purposes. If you want to run automate in HA mode, please [contact Chef Support](#).

To get started, go to marketplace and search for Chef Automate, or click one of following links:

- [AWS](#)
- [Azure](#)

The one-click deployments provision an instance from a templates that builds a VM with the recommended minimum 16GB of RAM, 4 CPUs and 80GB of disk space. If you want to set up custom security rules and other features, please see complete details on the [Chef Docs](#) page.

When the instance boots, your Chef Automate dashboard FQDN URL and credentials are automatically provided in the **Outputs** tab. Open a browser and use those credentials to get started. Once logged in, skip to **Step 3, Log into the Chef Automate Dashboard**, below for additional Chef Workstation configuration steps.

ChefInfraServer	<a href="https://ec2-54-208-25-47.compute-1.amazonaws.com/organizations/demo">https://ec2-54-208-25-47.compute-1.amazonaws.com/organizations/demo</a>	Chef Infra configuration management software eliminates manual efforts and ensures infrastructure remains consistent and compliant over its lifetime ? even in the most complex, heterogenous, and large-scale environments.
ChefAutomate	<a href="https://ec2-54-208-25-47.compute-1.amazonaws.com">https://ec2-54-208-25-47.compute-1.amazonaws.com</a>	Enterprise dashboard and analytics tool enabling cross-team collaboration with actionable insights for configuration and compliance and an auditable history of changes to environments.
Password	i-062863aef458ff76b	Password for Chef Automate
StarterkitPath	/home/ubuntu/starterkit-demo.zip	Path for the Starterkit zip file
Username	admin	Username for Chef Automate

## Deploy Chef Automate and Chef Infra Server Manually

If you want to deploy the stack manually, take the following steps.

Recommended minimum system requirements for Chef Automate running in production are 16GB of RAM, 4 CPUs and 80GB of disk space. For testing purposes, you can run it successfully with fewer resources in an on-prem or cloud Linux VM. The installation of Chef Automate also includes Chef Infra Server and can be fully deployed in about 20 minutes.

For this example, you can get away with fewer resources for a test environment, but such a set-up **should not** be used in production.

Chef Automate provides a dashboard for all your Chef-related activity and should be used with [Chef Workstation](#) and at least one Linux, Windows or Mac target nodes. Chef Workstation provides all the tools you need to interact with the Chef Server, including `chef`, `knife`, `inspec` and `hab`.

## 1. Install, Configure and Run Chef Automate

Before installing the Chef Automate stack, it's important your server has a fully qualified domain name (FQDN), such as **automate.chef.lab**, and a user other than root that has `sudo` privileges. The FQDN is what the nodes will use to resolve the server. In this example, I'm running a simple DNS server on my subnet so the Chef Automate/Chef Infra Server and all my target nodes can communicate. If you don't have DNS, you'll need to edit the `/etc/hosts` file on your server, workstation and nodes.

Having a FQDN is important because Chef Automate uses certificates that include your system's full hostname. So, for example, if your server's FQDN is merely "**automate**", the shared cert will only authorize to that short hostname. If other systems in your Chef ecosystem can't resolve that short name, most of your Chef activities won't work.

Check your system FQDN with the simple command:

```
$ hostname -f
```

If the result doesn't include your domain name, edit `/etc/hostname` and `/etc/hosts` as necessary, and reboot to ensure the FQDN is applied.

## 2. Deploy Chef Automate with `chef-automate`

We've made it possible to install all the components of a Chef Automate and Chef Infra Server using Chef itself by first downloading the standalone `chef-automate` application. It does all the installation work for you. Simply open a shell on your Linux server, use `curl` to get the latest version, and make it executable:

```
$ cd ~
```

```
$ curl https://packages.chef.io/files/current/latest/chef-automate-cli/chef-automate_linux_amd64.zip | gunzip - > chef-automate && chmod +x chef-automate
```

Before running the installation, add a couple settings to `/etc/sysctl.conf`:

```
$ sudo sysctl -w vm.max_map_count=262144
```

```
$ sudo sysctl -w vm.dirty_expire_centisecs=20000
```

If you ignore this step, the installer will fail during the pre-flight check. To ensure these settings persist across system reboots, make sure `vm.max_map_count=262144` and `vm.dirty_expire_centisecs=20000` are written and saved to `/etc/sysctl.conf`.

You're now ready to install. Simply execute the following command on the system:

```
$ sudo ./chef-automate deploy --product automate --product infra-server
```

You'll see the preflight checks and then the deployment will begin. Depending on your system's resources, this will take about 10 minutes and finish with a system health check. You can also check the status of the server at any time with a simple status command:

```
$ sudo chef-automate status
```

Note that `data_collection` is turned on and enabled by default, which means your target nodes can communicate with Chef Automate and show Chef Infra, InSpec and Habitat reporting. There's no need to patch the configuration to enable real-time reporting.

### 3. Log into the Chef Automate Dashboard

When the installation is complete, you'll drop back to the command prompt in your home directory, where you can access the newly created `~/automate-credentials.toml` file. It provides a URL and credentials to log into the Chef Automate dashboard.

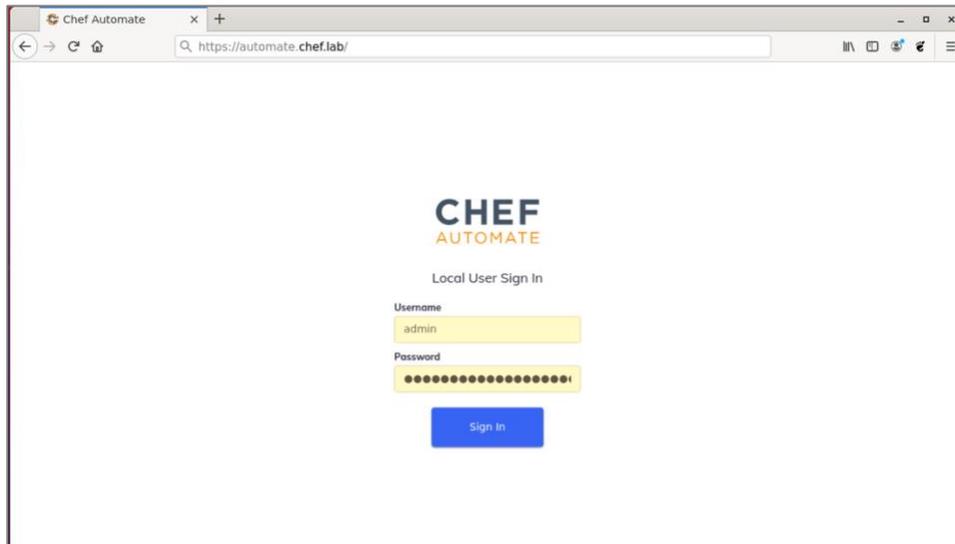
```
$ sudo vi ~/automate-credentials.toml

url = "https://automate.chef.lab"

username = "admin"

password = "5298c802960db6f1e339d97c8a33f81a"
```

Open a browser and use the provided URL and credentials to log into your Chef Automate server:



Before leaving the Chef Automate shell, create a Chef Infra Server user and organization. These enable Chef Workstation and your nodes to properly communicate:

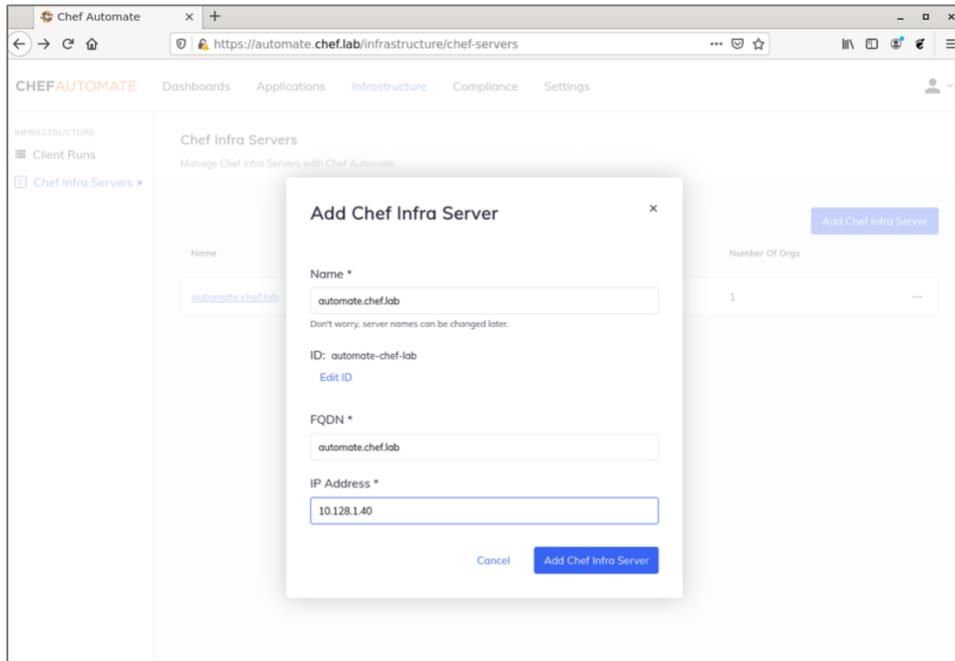
```
$ sudo chef-server-ctl user-create jtonello John Tonello
jtonello@chef.lab 'password' --filename jtonello.pem

$ sudo chef-server-ctl org-create lab 'Chef Lab' --association_user
jtonello --filename lab-validator.pem
```

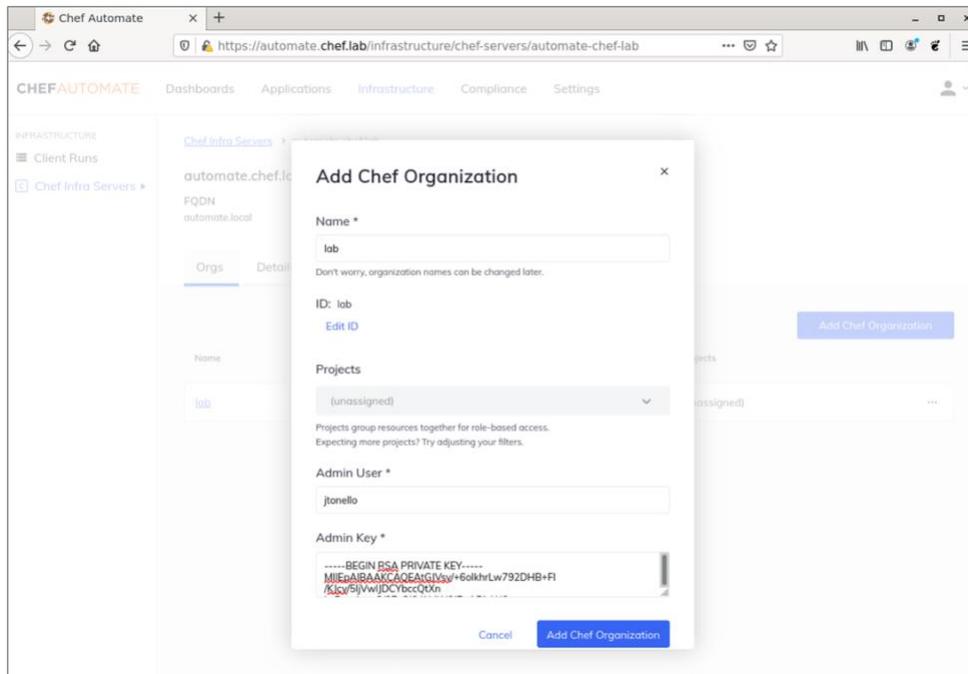
Use the `<username>.pem` file ("*jtonello.pem*" in this example) in the next step.

### 4. Add your Chef Infra Server to the Chef Automate Dashboard

Once you log in to your Automate Dashboard, click on the **Infrastructure** -> **Chef Infra Servers**, and click the **Add Server** button. Give the server a name, its FQDN (or something like `automate.local` since the server and Automate are running on the same machine) and provide the IP address.



With the server added, click on its link and click the **Add Chef Organization** button. Enter the same name you used in the CLI command above (“lab”). For the **Admin User**, enter the name you created above and copy the contents of your <username>.pem file into the **Admin Key** field. Click the **Add Chef Organization** button to save.



## 5. Configure Your Workstation to Communicate with Your New Server

Copy the <username>.pem file you created in **Step 4** from your Chef Automate server to the ~/ .chef directory on your Linux, Windows or Mac laptop machine running Chef Workstation. Here, I've opened a shell on the Chef Automate server and I'm copying it to a separate laptop. You can also copy and paste the contents to a new file with the same name on your workstation:

```
$ scp ~/jtonello.pem workstation.chef.lab:/home/<user>/.chef/
```

Finalize the configuration by opening a shell on your workstation machine and running the following knife command from your home directory:

```
$ cd ~  
$ knife configure init-config
```

When the command runs, it will prompt you to answer some questions and save the results in `~/.chef/credentials`, which will look something like the example below. Add a `cookbook_path` (or paths) that points to your workstation's Chef repo directory:

```
[default]  
client_name      = "jtonello"  
client_key       = "~/.chef/jtonello.pem"  
chef_server_url  = "https://automate.chef.lab/organizations/lab"  
validator_key    = "~/nonexist"  
cookbook_path    = ["~/chef-repo/cookbooks"]
```

Finally, you can confirm your set-up by running a few `knife` commands to fetch your new Chef Automate/Infra Server keys:

```
$ knife ssl fetch  
$ knife ssl check  
$ knife client list
```

That's it! You're now ready to bootstrap nodes, upload cookbooks and InSpec profiles, and see all your Chef activity in the Chef Automate dashboard.

### Next Steps

To learn more about Chef Automate, visit <https://www.chef.io/products/chef-automate> or the documentation site at <https://docs.chef.io/automate/>, which includes a Vagrant script to deploy the stack.

For more information on purchasing Chef products, please contact [sales@chef.io](mailto:sales@chef.io).

## ABOUT CHEF AND PROGRESS

Progress (NASDAQ: PRGS) provides the best products to develop, deploy and manage high-impact business applications. Acquired in October 2020, Chef extends Progress offerings in DevOps and DevSecOps, with market-leading, modern infrastructure, with compliance, and application by automation. With Progress, you can accelerate the creation and delivery of strategic business applications, automate the process which you configure, deploy and scale those apps, and make your critical data and content more accessible and secure— leading to competitive differentiation and business success. Over 1,700 independent software vendors, 100,000+ enterprise customers, and a three-million-strong developer community rely on Progress to power their applications. Learn about Progress at [www.progress.com](http://www.progress.com) or +1-800-477-6473.

<https://www.chef.io>