



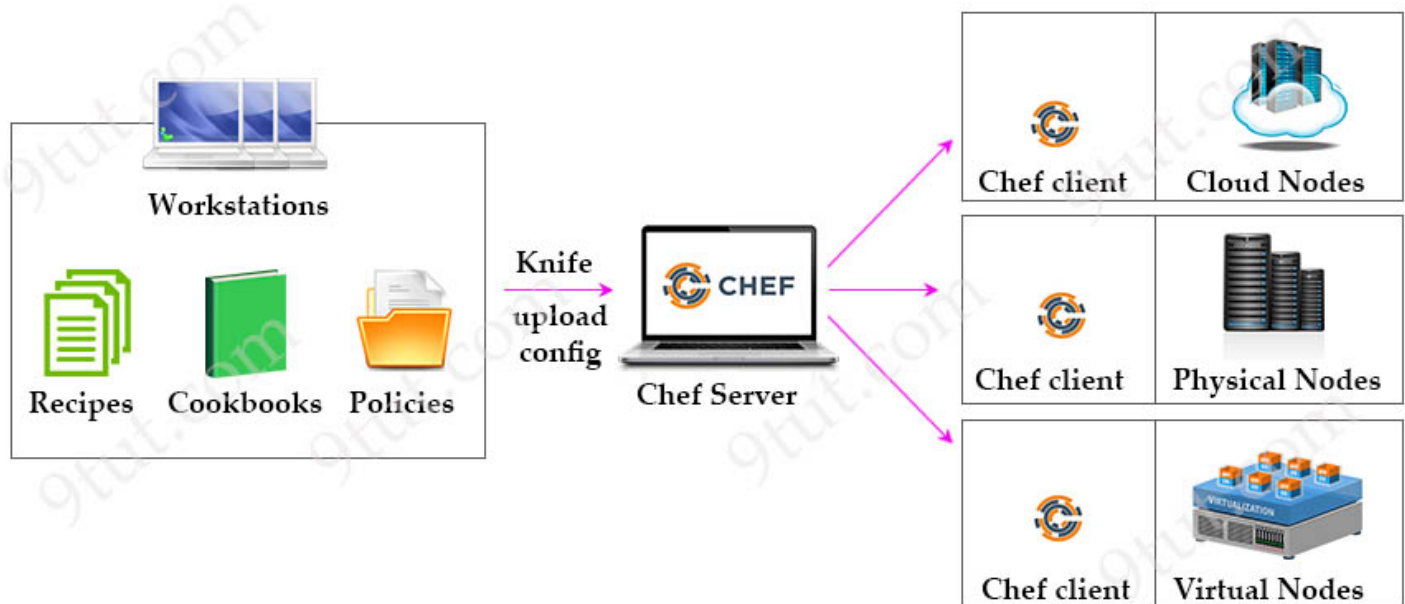
[Home](#) > Chef Tutorial

Chef Tutorial

December 3rd, 2020 [Go to comments](#)

Chef is an automation platform that configures and manages your network infrastructure. Chef transforms infrastructure into code. “Infrastructure into code” here means “deploy your code/application/configuration and policy” on many machines or instances automatically via your code.

As shown in the diagram below, there are three major Chef components:



+ **Workstations:** simply personal computers where all development configuration code is created, tested, and changed before uploading to the Chef Server. Each Chef workstation also has a command line tool called “Knife”, which will be used to upload configuration changes to the Chef Server.

Workstations are the place to write Recipes and Cookbooks:

++ **Recipes:** A Recipe is a collection of resources that describes a particular configuration or policy. It describes everything that is required to configure part of a system and in which order it is to be used. The user writes Recipes that describe how Chef manages applications and utilities (such as Apache HTTP Server, MySQL, or Hadoop) and how they are to be configured.

++ **Cookbooks:** Multiple Recipes can be grouped together to form a Cookbook. A Cookbook defines a scenario and contains everything that is required to support that scenario. A Cookbook also includes attributes, libraries, metadata, and other files that are necessary for supporting each configuration. Cookbooks are created using Ruby language and Domain Specific languages are used for specific resources.

+ **Chef Server:** The centralized store of our infrastructure’s configuration. The Chef server stores, manages and provides configuration to all nodes that make up the infrastructure.

+ **Nodes:** are the servers where your code needs to run. Chef server manages Nodes by **Chef client**, which is a software installed on each Node. Chef client retrieving configuration information from the Chef Server. Nodes can be a cloud-based/virtual/physical server in your own data center.

Any changes made to your infrastructure code must pass through the Chef server in order to be applied to nodes. Prior to accepting or pushing changes, the Chef server authenticates all communication via its REST API using public key encryption.

Chef client periodically pulls Chef server to see if there are any changes in cookbooks or settings. If there are changes then Chef server sends the latest configuration information to Chef client. Chef client applies these changes to nodes.



Sample Cookbook showing configuration of switch interface as Layer 3

```

cisco_interface 'Ethernet1/1' do
  action :create
  ipv4_address '1.1.1.1'
  ipv4_netmask_length 24
  ipv4_proxy_arp true
  ipv4_redirects true
  shutdown true
  switchport_mode 'disabled'
end

```

Or configure interface as Layer 2:

```

cisco_interface 'Ethernet1/2' do
  action :create
  access_vlan 20
  shutdown false
  switchport_mode 'access'
  switchport_vtp true
end

```

In summary, please remember the following important facts about Chef:

- + Use “pull” model (nodes are dynamically updated with the configurations that are present in the server)
- + Use TCP port 10002 (default command port) for configuration push jobs
- + Use Ruby for device configuration
- + Files needed for operation: Recipe, Cookbook...
- + Chef server works only in Linux/Unix but Chef client and Workstation can work in Windows as well.

We also made a comparison list of Ansible, Puppet and Chef automation tool here:

Criteria	Ansible	Puppet	Chef
Configuration Language	YAML, Python	+ Puppet DSL + Embedded Ruby	Ruby DSL, JSON
Architecture	Agentless (Client only)	Both (Agentless & Agent Based)	Agent Based (Client-Server)
Deployment Method	Push Model	Pull Model	Pull Model
Files created before Operation	Playbook	Manifest	Recipe
Availability	Ansible Primary Instance	Puppet Master	Chef Master
How to manage devices	Any device (can become controller)	Puppet Master	Chef Master
Installation	easiest	medium	hard
Transport Mechanism	SSH/NETCONF	REST	REST
Port used	TCP port 22	TCP port 8140	TCP port 10002
Initial Release	2012	2005	2009

[Comments \(3\)](#) Comments

1. realmadrid100001
March 15th, 2021

awesome, thanks for taking your time writing a short summary about chef

2. learner
July 28th, 2021

Good brief introduction with important information in an understandable way. Thanks for sharing.

3. 700-755 SBTO
February 27th, 2022

May you help to get exam materials for Small Business Technical Overview (SBTO)

Add a Comment

<input type="text"/>	Name
<div></div>	

Submit Comment

[Subscribe to comments feed](#)

[Puppet Tutorial](#) [Ansible Tutorial](#)

Premium Member Zone

Welcome [Gurjeet singh!](#)

- [Welcome Premium Member](#)
- [CCNA – New Questions Part 5](#)
- [CCNA – New Questions Part 6](#)
- [CCNA – New Questions Part 7](#)
- [CCNA – New Questions Part 8](#)
- [CCNA – New Questions Part 9](#)
- [Composite Quizzes](#)

- [Logout](#)

CCNA 200-301

- [Basic Questions](#)
- [Topology Architecture Questions](#)
- [Cloud & Virtualization Questions](#)
- [CDP & LLDP Questions](#)
- [Switch Questions](#)
- [VLAN & Trunking Questions](#)
- [VLAN & Trunking Questions 2](#)
- [STP & VTP Questions](#)
- [EtherChannel Questions](#)
- [TCP & UDP Questions](#)
- [IP Address & Subnetting Questions](#)
- [IP Routing Questions](#)
- [IP Routing Questions 2](#)
- [OSPF Questions](#)
- [OSPF Questions 2](#)
- [EIGRP Questions](#)
- [NAT Questions](#)
- [NTP Questions](#)
- [Syslog Questions](#)
- [HSRP Questions](#)
- [Access-list Questions](#)
- [AAA Questions](#)
- [Security Questions](#)
- [Security Questions 2](#)
- [DAI Questions](#)
- [IPv6 Questions](#)
- [DNS Questions](#)
- [QoS Questions](#)
- [Port Security Questions](#)
- [Wireless Questions](#)
- [Wireless Questions 2](#)
- [SDN Questions](#)
- [DNA Center Questions](#)
- [Drag Drop Questions](#)
- [Drag Drop Questions 2](#)

- [Drag Drop Questions 3](#)
- [VPN Questions](#)
- [DHCP Questions](#)
- [Automation Questions](#)
- [Miscellaneous Questions](#)
- [CCNA FAQs & Tips](#)
- [Share your new CCNA Experience](#)

CCNA Self-Study

- [Practice CCNA GNS3 Labs](#)
- [CCNA Knowledge](#)
- [CCNA Lab Challenges](#)
- [Puppet Tutorial](#)
- [Chef Tutorial](#)
- [Ansible Tutorial](#)
- [JSON Tutorial](#)
- [Layer 2 Threats and Security Features](#)
- [AAA TACACS+ and RADIUS Tutorial](#)
- [STP Root Port Election Tutorial](#)
- [GRE Tunnel Tutorial](#)
- [Basic MPLS Tutorial](#)
- [TCP and UDP Tutorial](#)
- [Border Gateway Protocol BGP Tutorial](#)
- [Point to Point Protocol \(PPP\) Tutorial](#)
- [WAN Tutorial](#)
- [DHCP Tutorial](#)
- [Simple Network Management Protocol SNMP Tutorial](#)
- [Syslog Tutorial](#)
- [Gateway Load Balancing Protocol GLBP Tutorial](#)
- [EtherChannel Tutorial](#)
- [Hot Standby Router Protocol HSRP Tutorial](#)
- [InterVLAN Routing Tutorial](#)
- [Cisco Command Line Interface CLI](#)
- [Cisco Router Boot Sequence Tutorial](#)
- [OSI Model Tutorial](#)
- [Subnetting Tutorial – Subnetting Made Easy](#)
- [Frame Relay Tutorial](#)
- [Wireless Tutorial](#)
- [Virtual Local Area Network VLAN Tutorial](#)
- [VLAN Trunking Protocol VTP Tutorial](#)
- [IPv6 Tutorial](#)
- [Rapid Spanning Tree Protocol RSTP Tutorial](#)
- [Spanning Tree Protocol STP Tutorial](#)
- [Network Address Translation NAT Tutorial](#)
- [Access List Tutorial](#)
- [RIP Tutorial](#)
- [EIGRP Tutorial](#)
- [OSPF Tutorial](#)

Network Resources

- [Free Router Simulators](#)
 - [CCNA Website](#)
 - [ENCOR Website](#)
 - [ENSDWI Website](#)
 - [ENARSI Website](#)
 - [DevNet Website](#)
 - [CCIE R&S Website](#)
 - [Security Website](#)
 - [Wireless Website](#)
 - [Design Website](#)
 - [Data Center Website](#)
 - [Service Provider Website](#)
 - [Collaboration Website](#)

[Top](#)



Copyright © 2021 CCNA Training

[Site Privacy Policy](#). Valid XHTML 1.1 and CSS 3.H