

## Chef Configuration Management

### Audience Course Chef Configuration Management

The course Chef Configuration Management is intended for system administrators and devops engineers who want to automate system administration and application deployment with Chef.

### Prerequisites Course Chef Configuration Management

General experience with system management in an IT infrastructure and knowledge of the basic principles of the Ruby script language is beneficial for the understanding.

### Realization Training Chef Configuration Management

The subject matter is discussed on the basis of presentation slides and demos. The theory is interchanged with exercises. The course material is in English. Course times are from 9.30 to 16.30.

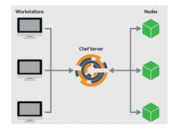
### Certification Chef Configuration Management

After successful completion of the course the participants receive an official certificate Chef Configuration Management.

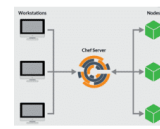
Duration: 2 days

Price: € 1499

[Open Schedule](#)



### Chef Configuration Management



## Content Course Chef Configuration Management

In the course Chef Configuration Management participants learn the open source Chef configuration management technology that is used to manage an infrastructure with physical or virtual servers. Chef uses Ruby to create recipes and cookbooks that automate manual and repetitive infrastructure management tasks. In DevOps Chef is used to deploy and manage servers and applications both locally and in the cloud.

### Chef Architecture

The course starts with a discussion of the components and topology of Chef and the basics of how Chef can be integrated with Cloud platforms. The Ruby Domain Specific Language (DSL) is also discussed here.

### Nodes and Roles

Subsequently, the different types of nodes that Chef knows are discussed, such as Physical Nodes, Cloud Nodes, Virtual Nodes and Network Nodes. The roles of Chef Workstation, Chef Server and Chef Client are also explained and the Ohai tool is treated.

### Building Blocks

Part of the program of the course are also the building blocks of Chef such as Recipes, Cookbooks, Packages, Services, Files and Templates. And Users and Groups and the use of Resources with their Attributes are discussed.

### Chef Cookbooks

Then the Chef Cookbooks are treated in more details. Attention is paid to the development of Recipes, the use of various attribute files, the use of libraries and the role of template variables.

### Extensions

A number of advanced applications of Chef are also discussed, such as the use of Custom Extensions, securing Data Bags, encryption of data and the storage of keys on nodes.

### Cloud Integration

Finally the course ends by explaining how Chef can be used in combination with Cloud platforms such as AWS and Azure.

## Modules Course Chef Configuration Management

Module 1 : Chef Intro	Module 2 : Chef Nodes	Module 3 : Building Blocks
What is Chef? Managing Infrastructure Deployment Automation Chef Components Chef Topology Cloud Integration Chef and AWS Chef and Azure Installing Chef Ruby DSL Language Ruby Gems	Physical Nodes Cloud Nodes Virtual Nodes Network Nodes Chef Workstation Chef Server Chef Client Ohai Tool Command Line Use ChefDK KitchenCI	Recipes Cookbooks Packages and Services Users and Groups Default Structure Resources Attributes Files Templates Metadata.rb Data Bags
Module 4 : Using Cookbooks	Module 5 : Advanced Chef	Module 6 : Chef and the Cloud
Multiple Attribute Files Developing Recipes Libraries Using Metadata Computed Configurations Template Resource Template Variables chef Command Using knife Testing Recipes Actions and Reports	Custom Extensions Cross Platform Cookbooks Building Resources Modifying Resources Securing Data Bags Encrypting Data Querying Data Storing Keys on Nodes Managing Users Distributing SSH Keys Adding Deployment Keys	AWS EC2 Bootstrapping Provisioning EC2 Instances knife-EC2 Plugin Configuration Options Configuring Database Host Configuring Web Server Chef Azure VM Extensions Azure Key Vault ARM Azure-cli and Powershell Azure Resource Manager Continuous Provisioning