Updated 04/11/2024

Sign up

Ansible training

2 days (14 hours)

Introducing Ansible

Thanks to its ease of use, Ansible will help you automate the setting up of complex infrastructures and the deployment of configurations and applications.

In this training course, designed for administrators and developers who want to get up and running straight away, you'll learn how to make intensive use of Ansible technology.

This 2-day training course will introduce you to the Ansible platform and its features. You'll learn how to simplify your environment by automating tasks in all your DevOps infrastructure projects. You'll also learn how to use best practices to master this technology.

As with all our training courses, this one will introduce you to the latest version of Ansible (at the time of writing: Ansible 2.16).

Objectives

- End-to-end control of your automation chain with Ansible
- All the concepts and key words of the language are covered and explained with concrete case studies
- The use of Ansible Tower's web interface (add-on module) through its open-source version AWX

Target audience

- Developers
- Architects
- System administrators

Prerequisites

- Have taken our Docker training course

Software prerequisites (in the case of in-house training)

- The latest OS updates (Linux, macOS or Windows under a subsystem for the Linux distribution: WSL)
- Python installed and updated
- Minimum configuration
  - 2GB RAM
  - 20GB hard disk
  - 64-bit processor

Pre-course reading recommendations

- We recommend Stéphane Robert's blog on mastering Ansible, where you can learn the basics of this management tool.
- The Cherry Servers blog, where the main concepts of Ansible and DevOps are discussed
- Ansible documentation to review ;)
- The official glossary to avoid misunderstandings

Ansible Training Program

Part 1: Application containers with Docker

- What is a container?
- How a modern OS works: Kernel Land and User Land
- The differences between containers and virtual machines
- System container, application container
- Introducing the Linux container concept
- Linux container use cases
- Discover Docker
  - Is Docker a misnomer?
  - Introduction to Docker and its architecture
  - Advantages and disadvantages of Docker
- Other container managers

TP 1: Getting started with Docker

- Installing Docker
- Launching a container with docker run
- Interactive mode
- Interact with a container from the host (exec, inspect, logs...)
- Administration, cleaning of launched containers
- Building our own images
- What is a Docker image?

- Creating an image from a Dockerfile
- The public registry: Docker Hub
- Use a private register

TP 2: Our first image

- A simple application
- A naive Dockerfile
- Image construction and upload
- Dockerfile enhancement: multistage build
- Latest is not a
- Advanced concepts
- Data persistence
- The Docker network model
- What is a Cloud Ready application?
  - The 12 factors
  - What if my application isn't Cloud Ready?
- CI/CD pipeline automation
- Product lifecycle : Integration, Delivery, Deployment
- Overview of CI/CD platforms
- From Continuous Integration to Continuous Delivery

TP 3: Setting up a CI/CD pipeline

- Our project on Gitlab
- Each stage of the pipeline rotates... in a container!
- Unit testing: before or after the build?
- Kaniko: building a dep image
- Optimization: Building a toolchain image

Part 2: Orchestrating containers: Kubernetes

- Why do you need an orchestrator?
- High availability
- Service disovery
- Rolling update, rollbacks
- Policy
- ... And more!
- Kubernetes: the undisputed leader among container orchestrators
- Architecture
- Key concept: the reconciliation loop

WP 4: Minimalist Kubernetes on our workstation

- Minikube, k3s, microk8s, k0s, which to choose?
- Installation
- Checking the health of Kubernetes
- First orders
- Essential addons

- Deploying applications on Kubernetes
- Overview of resources
- Defining a resource: the manifest

TP 5: Progressive writing of our application's manifests

- Manifest format
- The basic unit: Pod
- Add high availability: ReplicaSet
- Add lifecycle management: Deployment
- Exposing the application: Service
- Abstractions on manifests: Helm and Kustomize
  - Helm, Kubernetes' package manager
  - Kustomize, YAML generation
  - Really competitors?
- Continuous deployment: GitOps
- What is GitOps?
- Argo CD

TP 6: Continuous deployment of our application

- A new infrastructure depot
- Argo CD installation
- Monitor the code repository
- Stateful application management with operators
- What is the operator pattern?
- Deploying stateful on Kubernetes or not: a strategic choice

Part 3: When a container is impossible: Ansible

- When is deploying on Kube unsuitable?
- "Kubernetes supports stateful workloads, I don't"
- Greenfield vs Legacy
- Introducing Ansible
- The "provisioning / configuration management / app deployment" pyramid
- Ansible is just a wrapper for SSH!
- Where does Ansible fit into this ecosystem?

TP 7: Set-up, first overview

- A detour into Python project configuration
- Connecting with SSH: generating a keypair, using the ssh agent
- First commands, `ansible -m ping` ...
- Playbooks
- Basic syntax
- Quick overview of the `builtin` collection
- Templating, Jinja2
- register, when...
- Organizing your variables

TP 8: Writing the first playbooks

- Hello, World!
- Installing Nginx
- Deployment of a typical stack
- Roles
  - What is a role?
  - Golden rules and best practices

TP 9: Let's clean up the previous TP using the roles

- A naive first pass
- Drawing on the state of the art
- A role repository usable by ansible-galaxy

Complementary module +3 day : Ansible Advanced Training

DevOps: an introduction

- History of DevOps
- Industrialization: a necessity for operations
- DevOps tools
- Situating Ansible within these tools

Ansible

- Why Ansible?
- SSH operation & link
- Installation

Practical work

Installing Ansible and preparing for use

Ad Hoc parameterization and control

- SSH settings
- Inventory preparation
- Ad-hoc" commands
- Module presentation

Practical work

Using Ansible: file transfer and commands on nodes.

Deployment and organization: Playbooks

- Introducing Git
- Introducing Playbooks
- Introducing YAML
- Variables and facts
- Loops and conditions
- Roles and inclusions
- Best practices for writing Playbooks

Practical work

Writing Playbooks for application deployment, managing a complete environment with Ansible

Advanced concepts

- Basic modules
- Module architecture and behavior
- Ansible-Galaxy

Practical work

Handling external modules

Additional module +1 day : Ansible training

- Ansible presentation, origin and terminology
- Different installation techniques using a Python virtualenv, packages and source code. Superior version 2.9x
- Ad-hoc control practice
- Controls and modules
- Create your own filters and modules
- Static and dynamic inventories
- Understanding host_vars and group_vars
- Exercises with Jinja2
- Handler and notify support
- Roles, collection and multi-os access
- Ansible-galaxy and automation hub
- Debugging and fine-tuning playbooks
- Script performance and optimization
- Secrets and Ansible Vault
- Ansible AWX , job and workflow management

Companies concerned

This training course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.