

**Introductory and Intermediate Ansible Configuration and Administration**

**Course Number:** ANS-104
**Duration:** 5 days

**Overview**

Ansible is a powerful open-source engine for automating configuration management, software provisioning, and deploying applications. This Introductory and Intermediate Ansible training course teaches attendees Ansible basics including how to build playbooks, leverage modules, and configure roles. Students then take their skills to the next level with advanced topics including playbook construction, module utilization, and node management, and more. In addition, this course covers the Ansible Tower.

**Prerequisites**

It is assumed that participants are working systems administrators, developers, and/or testers with some very basic scripting knowledge (bash, ksh, Perl or Python) and have an understanding of fundamental system utilities/commands on Linux and Windows systems.

**Materials**

All Ansible training students receive comprehensive courseware covering all topics in the course.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computer for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will be fine.

**Objectives**

* Install and configure Ansible control-node and managed-node(s)
* Create and run playbooks
* Use standard Ansible modules
* Create custom modules
* Create and modify hosts (inventory) definitions
* Create and use roles
* Design a 'best practice' strategy for using Ansible capabilities
* Explore Ansible Tower as a solution for graphically managing Ansible enterprise-wide
* Perform advanced configuration for an Ansible control node and managed nodes
* Use conditionals, loops, filters, and loopbacks in playbooks
* Control ad-hoc and playbook output with callback plugins
* Create and implement dynamic inventory definitions
* Implement reusable roles capabilities and dependencies
* Control Ansible Core operations with run-time parameters
* Create templates to write an Ansible custom module
* Add documentation to an Ansible customized module
* Create and use local fact variables
* Implement local fact variables on managed nodes
* Write playbooks to collect data from managed nodes

**Outline**

* Introduction
	+ Software development (cycle) considerations
	+ Strengths and weaknesses of Ansible
	+ Ansible terminology
	+ Ansible required environment setup
* Initial Setup and Configuration
	+ Ansible Core installation methods: rpm, pip, and Tower
	+ Ansible component locations
	+ Ansible documentation
	+ Ansible configuration file
	+ Inventory (hosts) file (Linux and Windows nodes)
	+ Gathering system facts (methods)
	+ Using system facts
* Ansible Playbooks
	+ Layout of playbook sections (directives, variable definitions, tasks, handlers, module modifiers)
	+ Variable creation and usage
	+ Using modules in playbooks
	+ Controls: loops, conditionals, tags, notifications, plugins, filters, and lookups
	+ Running as root
* Ansible Modules
	+ The layout of an Ansible module
	+ Linux modules:  file, yum, systemd, cron, user, shell, filesystem
	+ Using the Windows-specific (win\_) modules
	+ Common Linux and Windows modules
* Roles in Ansible
	+ Overview of a role
	+ Creating a role (structure)
	+ Using (a) role(s)
	+ Packaging up a role
	+ Ansible Galaxy - capabilities and usage with roles
	+ Git repo (role) repository (creation)
	+ Local access of a Git repo role repository
	+ Remote access of a Git repo role repository
* Managed Nodes
	+ Requirements for a Linux managed node
	+ Inventory and access control information on the control node
	+ Setup of a Linux managed node
	+ Requirements for a Windows managed node
	+ Setup of a Windows managed node
	+ Inventory and access control information on the control node
* Ansible Tower
	+ Comparison of Ansible Tower and Ansible AWX
	+ General layout of the Ansible Tower
	+ Installation of Ansible Tower (and Ansible AWX)
	+ Using the Ansible Tower / AWX (dashboard)
	+ Define Ansible Tower / AWX user accounts
	+ Create inventories of systems (with credentials)
	+ Create projects and job templates
	+ Job scheduling (launch forms), status, and tracking
	+ Watching and chaining (multi-) playbooks (workflows)
	+ Controlling and viewing logs and audit trails
	+ Notifications
* Control Code and Managed Node Required Configuration
	+ Ansible components review
	+ Running Ansible in escalated privilege mode
	+ SSH key setup
	+ Ansible common account creation, setup, and features
	+ Required components for a Linux/Unix/Windows managed node
* Advanced Control Node Setup and Configuration
	+ Ansible Core installation methods: rpm, pip, and Tower (changes)
	+ Ansible command and module documentation
	+ Inventory hosts file contents (Linux/Unix, Windows, and network device nodes)
	+ Dynamic inventory hosts file contents and usage
	+ Ansible configuration file (locations and parameter definitions)
	+ Special use configuration parameter definitions
	+ Methods to control gathering of system facts
	+ Creating and using local system facts
	+ Understanding the hostvars dictionary usage
	+ Variable creation and usage
* Ansible Playbooks
	+ Layout of a playbook: directives, tasks and arguments, module modifiers
	+ Installing and using ansible-lint
	+ Playbook: loops, conditionals, tags, notifications, plugins, filters, and lookups
	+ Controls: output formatting and inclusion with callbacks
	+ Advanced Jinja 2 template usage
	+ Using “inner” and “outer” playbooks
* Ansible Modules/Tasks
	+ Including external tasks in a playbook
	+ Controls over node and task scheduling
	+ Configuration parameters that control rolling updates
	+ Using Ansible playbook debugging capabilities
	+ Writing a customized module and embedded documentation
* Roles in Ansible
	+ review of role creation / structure, packaging, and usage
	+ importing role definitions (on a control node)
	+ role dependencies definitions and usager
* Collections in Ansible
	+ Collection creation and structure (components)
	+ Using a role inside a collection
	+ Using a module inside of a collection
	+ Using a playbook inside of a collection
* Conclusion