+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 08/19/2026

Sign Up

PowerShell DSC Training

3 days (21 hours)

Overview

PowerShell Desired State Configuration (DSC) enables you to automate the configuration of Windows environments in a declarative, repeatable, and controlled manner. With DSC resources, MOF files, the Local Configuration Manager, and drift management, you can ensure reliable server deployment and operation.

Our PowerShell DSC training will enable you to develop an Infrastructure as Code approach tailored to Microsoft environments, allowing you to define, apply, and maintain consistent configurations across multiple machines.

You'll learn how to design PowerShell configurations, use standard and community DSC resources, compile configuration files, and manage their local or remote application using PowerShell Remoting mechanisms.

By the end of the training, you will be able to manage dependencies between components, configure nodes, verify compliance, and correct configuration discrepancies in a reliable and reproducible manner.

This training also covers meta-configurations, how the Local Configuration Manager works, securing sensitive data, troubleshooting errors, and best practices for automation in a DevOps pipeline.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Understand the principles of Desired State Configuration and declarative infrastructure.

- Create and compile PowerShell DSC configurations for Windows servers.
- Use DSC resources, dependencies, and node configuration data.
- Apply, monitor, and correct compliance with a target configuration.
- Configure the Local Configuration Manager and troubleshoot DSC deployments.
- Structure reusable, secure configurations that can be integrated with DevOps practices.

Target Audience

- Windows System Administrator
- DevOps Engineer
- Infrastructure Engineer
- PowerShell developer
- Microsoft System Architect

Prerequisites

- Proficiency in the fundamentals of Windows Server and its administration.
- Proficiency in PowerShell scripts, variables, functions, and modules.
- Basic knowledge of WinRM and remote administration.
- Experience deploying or configuring Windows services.

Technical Prerequisites

- A computer running Windows 10 or Windows 11, with local administrator privileges.
- Test environment available on Windows Server 2019, Windows Server 2022, or an equivalent virtual machine.
- Windows PowerShell 5.1 and Visual Studio Code installed.
- Stable Wi-Fi or Ethernet connection for module installation and remote work.

Our PowerShell DSC Training Course Outline

[Day 1 - Morning]

Fundamentals of Declarative Configuration

- The role of PowerShell DSC in an Infrastructure as Code approach.
- Principles of declarative configuration, idempotence, compliance, and drift correction.
- Overview of DSC versions and identification of use cases for Windows PowerShell DSC, PSDesiredStateConfiguration, and modern DSC.
- Historical architecture: nodes, resources, configuration documents, MOF compilation, and LCM engine.
- Exploring the cmdlets, modules, and resources available in the PowerShell environment.
- Hands-on workshop: preparing the workstation, inventorying DSC resources, and verifying the compliance status of a local workstation.

[Day 1 - Afternoon]

Write and apply an initial configuration

- Syntax of a PowerShell configuration, Node blocks, and resource declarations.
- Importing modules with Import-DscResource and using PSDscResources.
- Managing the desired state using the Ensure property, paths, contents, and attributes.
- Compiling configurations into MOF files and reading their structure.
- Controlled application with Start-DscConfiguration, execution options, verbosity, and waiting.
- Hands-on workshop: Create, compile, and apply a configuration that ensures the presence of files, folders, and environment variables.

[Day 2 - Morning]

Modeling Reusable Configurations

- Organizing a DSC project: scripts, modules, data, environments, and naming conventions.
- Configuring settings with `param`, variables, and configuration data ConfigurationData.
- Targeting multiple nodes and distinguishing between common and specific parameters.
- Managing runtime dependencies with DependsOn and the deployment order of components.
- Using File, Service, Registry, WindowsFeature, and Package resources.
- Hands-on workshop: Building a configurable setup to prepare an application server with roles, services, and system settings.

[Day 2 - Afternoon]

Managing Nodes and the Local Configuration Manager

- The role of the Local Configuration Manager in applying, monitoring, and remediating configurations.
- Reading and defining a meta-configuration using the ApplyOnly, ApplyAndMonitor, and ApplyAndAutoCorrect.
- Settings for frequency, restart, module management, and behavior after an error.
- Local and remote administration using WinRM, CIM sessions, and node management.
- Status monitoring using Get-DscConfiguration, Test-DscConfiguration, and Get-DscLocalConfigurationManager.
- Hands-on Workshop: Configure the LCM of a test node, trigger a controlled drift, and then verify whether it is detected and remediated.

[Day 3 - Morning]

Securing and Ensuring the Reliability of DSC Deployments

- Managing sensitive information in configurations, limitations of MOF files, and encryption principles.
- Using certificates to protect credentials and confidential data.
- Best practices for managing service accounts, secrets, permissions, and remote access.
- Diagnosing a DSC application: logs, detailed output, resource errors, and dependency analysis.
- Strategies for testing, validating, and versioning configurations in a Git repository.
- Hands-on workshop: Securing a configuration using a service account and analyzing a deployment failure based on DSC logs.

[Day 3 - Afternoon]

Industrializing DSC in a DevOps Approach

- Choosing between one-time execution, compliance checking, and automatic remediation based on operational contexts.
- Directly invoking resources with Invoke-DscResource and the Get, Test, Set approach.
- Interoperability with modern DSC, YAML and JSON files, and reuse of existing PowerShell resources.
- Integration into a CI/CD pipeline: syntax validation, testing, packaging, and deployment between environments.
- Designing a strategy for phased deployment, rollback, and compliance monitoring.
- Hands-on workshop: complete a mini-project to fully configure a Windows web server, versioned, and ready for integration into a DevOps pipeline.

Target Audience

This training is intended for both individuals and companies—large or small—seeking to train their teams in new, advanced IT technologies or to acquire specific industry knowledge or modern methodologies.

Pre-training Assessment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Hands-On Training: 60% Practical, 40% Theoretical. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor—supported by examples and reflection sessions—and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

A certificate will be issued to each trainee who has completed the entire training program.