

# Puppet Configuration and Management

## Advanced Concepts and Techniques

### Course Description

This course presents the system management and configuration tool **Puppet Enterprise**, **building upon the information contained and skills learned in the Basic / Intermediate course, to learn advanced concepts and utilize advanced capabilities.**

### Course Objectives

At the **completion** of the **course**, **participants** should be **able to** :

- create manifests to install and configure **Puppet** servers and agents
- configure complex systems infrastructure using optimized declarations
- learn more capabilities of classes and modules (in **PE 4**)
- examine the usage of additional **Hiera** backends
- use (and extend) **Factor** (external) facts
- write and deploy customized **Ruby** facts
- module-specific facts
- extend **Puppet environments** with **r10k**
- new data structures and type manipulation statements
- track resource changes with auditing
- introduce roles and profiles

### Implementation

Each participant will have a **Linux** virtual server for a **Puppet Enterprise 4** installation, resource definition writing and application. Depending upon the available resources on the host system (**Windows** or **Mac OS X**), a second **CentOS 7, Solaris 10, or Windows** virtual system will be setup as the **Puppet** agent. The system will also be used as a **masterless Puppet configuration**.

### Audience Prerequisites

**Participants** are **assumed** to have (either) **attended the Puppet Configuration and Management (Basic / Intermediate Concepts and Techniques) course**, or have **equivalent time as a working system administrator and as a Puppet Master**.

# Puppet Configuration and Management

## Advanced Concepts and Techniques

### Course Content

- **Configuration Management**
  - extending **Facter** (external and internal facts)
  - directory locations
  - Ruby** defined
  - usage with modules
  - agent defined facts
- **Puppet Resource and Module Definitions**
  - advanced resource definition types
  - (new) conditional and loop constructs
  - implementing new resource type definitions
  - additional resource relationships and metaparameters
  - defining and tracking resource changes with auditing
  - automating the inclusion of class definitions
  - more on inheritance with classes (**hiera\_include**)
  - defining a module repository
- **File Access and Content**
  - using **Puppet URLs**
  - file()** function and file resource type definition
  - using templates
    - ERB** (embedded **Ruby** based)
    - EPP** (embedded **Puppet** based)
- **Environments**
  - best practice usage of environment definitions
  - managing the use of environments with **r10k**
- **Extending Puppet with Hiera**
  - Hiera**
  - additional backend types (**file**)
  - parameterized classes with automated data binding (**ADB**)
  - interaction with **Facter**
  - defining and using roles and profiles

### Duration

This course normally requires **two (2) days**, 50% lecture, with 50% hands-on commands and lab exercises