Puppet Essentials Training

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com

Duration: **4 Day**

This course provides the necessary skills and techniques to install Puppet Master and Agent in order to carry out configuration management effectively.

The course is presented as a mixture of lectures and hands-on exercises.

Exercises are provided for both the Enterprise and Open Source versions of Puppet.

Prerequisites

Before attending this course, students must have:

- Experience in using the command line
- Some familiarity with Linux (or similar) concepts
- Experience with a text editor such as vi
- A basic understanding of services and host file entries

What You Will Learn

Delegates will learn how to:

- Install and configure Puppet Enterprise
- Install and configure Puppet Agent
- Describe the key components of Puppet including Hiera and Facter
- Write classes comprising desired resources for a node
- Test their Puppet configurations
- Use Puppet to configure multiple web servers

Training Outline

Introduction to Configuration Management

- Why do we need Configuration Management?
- What are the components of Configuration Management?
- What are the benefits of Configuration Management?
- What is the Configuration Management life cycle?
- What are the common tools for Configuration Management?

Puppet Overview

- Puppet Enterprise stack and features
- Model based approach
- Composable configurations
- Lifecycle of a puppet run
- Puppet component roles (master / agent)

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Master / agent architecture

Puppet Components

- Describing and maintaining state
- Infrastructure as code
- Idempotency
- Resources
- Live management
- Facter, Hiera and Puppet forge

Managing assets with Git

- Source Control
- Version Control
- Centralised and Distributed Version Control Systems
- GIT as a DVCS
- GIT Basics
- Choosing a DVCS
- The SSH Protocol

Resources and Providers

- Puppet resources
- Resource declarations
- Declarative Modelling Language
- Resource Abstraction Layer
- Providers

Modules and Classes

- Puppet Classes
- Puppet Modules
- Module Directory Structure
- Namespaces and Auto-loading
- Defining and Declaring Classes
- Testing Classes

Using Puppet Without a Master

- Puppet Usage Patterns
- Masterless Puppet - in short
- Benefits of going 'Masterless'
- What would you have to do?
- Problems with going 'Masterless'

Puppet Essentials Training

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com

Duration: **4 Day**

Deploying Your Code to the Puppet Master

- A Git to Puppet Workflow
- Git and Puppet Master
- Single or Multiple Repositories?
- Managing Multiple Masters

Testing Puppet

- Syntax checking
- The Smoke test
- Simulating change with Puppet
- Virtualisation
- Module testing frameworks
- Unit testing modules
- Test Driven Development - TDD

Using Puppet Forge

- Puppet Forge
- Using the Puppet Forge online
- The puppet module tool - PMT
- Search for modules
- Install a module from Puppet Forge
- Uninstall a module
- Publish a module to the Puppet Forge

Classification

- What is Classification?
- Node Definitions: site.pp
- Regular Expressions in Node Definitions: site.pp
- Console Classification
- Console Classification - Node Definitions
- Console Classification - Node Groups

Environments

- What are Environments?
- Environment Directory Structure
- Configure Puppet Master to use Directory Environments
- Global Settings for Configuring Environments
- Assign Nodes to their environments
- Preconfigured Node Groups
- Environment-based Workflow

Puppet Essentials Training

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Duration: **4 Day**

Live management

- What is Live Management?
- Inspect node Resources
- Control Puppet

ERB templates

- Use cases for ERB Templates
- Using templates
- Using Puppet variables
- Defining Puppet variables
- ERB Conditions

Resources in depth

- Resource type origins
- Resource type categories
- Meta resource types
- Resource type documentation
- Metaparameters
- title and namevar

Resource relationships

- Resource relationships
- Dependency management
- Implicit dependencies

DSL - Domain Specific Language

- Using Puppet variables (facts)
- Defining local variables
- Using variables
- Variable scope
- Strings and arrays
- Conditional expressions: if, selectors, case, unless
- Statements and functions

Parametrised types

- Parameterised Classes and Inheritance
- Defining a Parameterised Class

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Declaring a Parameterised Class
- Assigning Parameters in the PE Console
- Class Inheritance - base class declaration
- Class Inheritance - sub class declaration
- Class Inheritance - Best Practice

Defined resource types

- Defining and Declaring a Defined Type
- Module Directory Structure

Hiera

- What is Hiera?
- Why use Hiera?
- Configuring Hiera
- Yaml file locations
- Sample data files
- Automatic data bindings
- Explicit Hiera lookup
- Testing Hiera

PuppetDB

- What is PuppetDB?
- What is stored in PuppetDB?
- How is PuppetDB accessed?
- Example queries
- Dashboards
- PuppetDBquery module

Troubleshooting

- Troubleshooting Puppet classes and Modules
- Troubleshooting Puppet Installations

Puppet Best Practices

- Using Modules
- Version Control
- Naming Conventions
- Style
- Classes Vs Defined Types
- Roles and Profiles

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

Other Configuration Management Tools

- Managing AWS services with the AWS CLI
- Configuring Servers using Kickstart
- Provisioning Infrastructure with Foreman and Puppet
- Preparing operating systems using Cobbler