Puppet

将基础设施作为代码进行保存与版本化

无论是在笔记本上的开发环境，还是在生产环境上，开发人员和运维人员都能够使用相同的清单对系统进行管理

将配置作为代码处理，系统管理员就能够为开发人员提供独占的测试环境

甚至可以将Puppet代码交付给审记，如今有许多审记都接收Puppet清单，以进行一致性验证

能够将Puppet代码签入到某个共享的版本控制工具中，这将为你的基础设施提供一个可控的历史记录。

A Configuration Management Tool

A framework for Systems Automation

A Declarative Domain Specific Language (DSL)

Configuration Management advantages

Infrastructure as Code: Track, Test, Deploy, Reproduce, Scale

Code commits log shows the history of change on the infrastructure

Reproducible setups: Do once, repeat forever

Scale quickly: Done for one, use on many

Coherent and consistent server setups

Aligned Environments for devel, test, qa, prod nodes

Puppet Tutorial

By Alessandro Franceschi

Example42

11-07-2015

Puppet Essentials - Overview

Introduction to Puppet

Configuration management tools

Puppet Ecosystem and related software

What is Puppet

Software related to Puppet:

Facter - Complementary tool to retrieve system's data

MCollective - Infrastructure Orchestration framework

Hiera - Key-value lookup tool where Puppet data can be placed

PuppetDB - Stores all the data generated by Puppet

Puppet DashBoard - A Puppet Web frontend and External Node Classifier (ENC)

The Foreman - A well-known third party provisioning tool and Puppet ENC

Geppetto - A Puppet IDE based on Eclipse

Puppet Tutorial

By Alessandro Franceschi

Example42

11-07-2015

Puppet Essentials - Overview

Introduction to Puppet

Configuration management tools

Puppet Ecosystem and related software

What is Puppet

Installation

Debian, Ubuntu

Available by default

apt-get install puppet # On clients (nodes)

apt-get install puppetmaster # On server (master)

Puppet Language

A Declarative Domain Specific Language (DSL)

It defines STATES (Not procedures)

Puppet code is written in manifests (files with .pp extension)

In the code we declare resources that affect elements of the system (files, packages, services ...)

Resources are grouped in classes which may expose parameters that affect their behavior.

Classes and configuration files are organized in modules.

Nodes classification

When clients connect, the Puppet Master generates a catalog with the list of of the resources that clients have to apply locally.

The Puppet Master has to classify nodes and define for each of them:

The classes to include

The parameters to pass

The Puppet environment to use

The catalog is generated by the Master according to the logic of our Puppet code and data.

In our code we can define our variables and use other ones that may come from different sources:

facts generated directly by the client

parameters obtained from node's classification

Puppet internal variables

Resource Types (Types)

file { 'motd':

path => '/etc/motd',

content => 'Tomorrow is another day',

}

Why Puppet

As system administrators acquire more and more systems to manage, automation of mundane tasks is increasingly important.

use Puppet to manage a machine throughout its lifecycle — from initial installation, to ongoing upgrades, and finally to end-of-life

Vagrant

创建和部署虚拟化开发环境。它使用Oracle的开源VirtualBox虚拟化系统，使用 Chef创建自动化虚拟环境

vagrant的打包功能类似Vmware的克隆，打包之后可以拿给其他人去用，统一的平台，统一的软件结构，作为开发测试的一个公共平台实在是极好的~

唯一的不同点就是Vmware的克隆出来一个虚机很大很大，而vagrant重新封包依旧是小巧玲珑~