

OPS635 Lab 2

System Configuration Management using Puppet

Overview

In an enterprise environment, the existence of different hardware and Operating Systems running on production servers is very common. You may find the same software running on different OS and even different hardware platforms. To manage the software configuration on different platforms by using a script, your script must be aware and account for their differences. Your scripts must be written in way that it knows how to query the operating environment and execute the appropriate installation and configuration actions based on the operating environment it is running on. There are commercial and open source configuration management systems which are aware and know how to handle such differences in operating platforms. "Puppet" is one of those CMS that we are going to study for this lab. In this lab, we will start by trying to port the script that you have written for configuring Nagios on CentOS7 to Debian/Ubuntu (or any other Linux distributions of your choice except RHEL and Fedora). Secondly, we will convert your original Nagios installation and configuration script into a Puppet manifest, test it in your testing environment and then deploy it to your production environment (i.e. your Seneca VM).

Tasks

1. Port your original Nagios installation and configuration script from CentOS 7 to another Linux Distribution of your choice (excluding RHEL, Fedora)
2. Install and configure Puppet on a local VM running CentOS 7
3. Install and configure Puppet on a local VM running another Linux Distribution of your choice (excluding RHEL, Fedora)
4. Convert your original Nagios installation and configuration script to a "Puppet Manifest"
5. Test your "Puppet Manifest" on both of your local VMs

Investigation 1: Porting an Installation Script to a Different Distribution

- Install a new VM using a different distribution of your choice.
 - Ensure it is fully updated.
 - Provide it a static address and hostname of your choice (this lab will refer to it as distro-blank), and add it to your DNS domain
 - As with your first 'blank' VM, we will try to avoid making changes to this VM, as we will clone it for several purposes.

- Clone the vm and provide the clone with a new name and address of your choice (this lab will refer to it as distro-clone).
- Convert your nagios installation script from lab 1 to work for the distribution you chose.
 - Note that you are not expected to spend an inordinate amount of time on this. The intent is for you to see how complicated it can be to manually modify even a simple installation process to work on different distributions. If you spend more than an hour and the script is still not working, just move on.

Investigation 2: Basic Puppet Installation

- Clone your Centos blank Vm to create two new Vms (puppet-master and puppet-client).
 - Set the hostnames of both, provide them with static addresses, and add them to your DNS service.
- Configure your puppet-master as the puppet master, and configure puppet-client and distro-clone as puppet clients.
- Write a puppet manifest on puppet-master to make a simple change (e.g. installing bind-utils) on both clients. Ensure it works before continuing.

Investigation 3: Using a Puppet Manifest

- Convert your nagios installation script into a puppet manifest that will configure both puppet clients (puppet-client and distro-clone) as nagios servers checking the same things as in lab 1.
- Ensure that even if you manually break your machines (e.g. remove a required package), they will automatically be fixed by the puppet-master.

Submission

Demonstrate your puppet manifest working in class, and upload the manifest to Blackboard.