Tech Talk (Report): 'Puppet'

Team: Abhimanyu Jataria (ajatari), Ankur Garg (agarg12), Atit Shetty (akshetty), Debosmita Das (ddas5)

Introduction – What is Puppet?

Puppet is a configuration management tool developed by Puppet Labs to automate infrastructure management and configuration. Puppet is a very powerful tool which helps in the concept of Infrastructure as code. This tool is written in Ruby DSL language that helps in converting a complete infrastructure in code format, which can be easily managed and configured.

Puppet Architecture

Puppet follows server-client model, where one node in the cluster serves as the master and stores all configurations of the clients and other nodes serve as agents and actually set up these configuration on the client nodes.

Puppet Master

Puppet Master is the key mechanism which handles all the configuration related stuff. It applies the configuration to nodes using the Puppet agent. It stores the configuration for various nodes written in DSL Language (based on Ruby)

Puppet Agent

Puppet Agents are the actual working machines which are managed by the Puppet master. They have the Puppet agent daemon service running inside them. These agents pull the latest configuration from the puppet master and check if there are any differences and then apply those to the node and then report back to the puppet master.

How does it Work?

- In Puppet, the first thing what the Puppet master does is to collect the details of the target machine. Using the 'factor' which is present on all Puppet nodes it gets all the machine level configuration details. These details are collected and sent back to the Puppet master.
- Then the puppet master compares the retrieved configuration with defined configuration details, and with the defined configuration it creates a catalog and sends it to the targeted Puppet agents.
- The Puppet agent then applies those configurations to get the system into a desired state.
- Finally, once one has the target node in a desired state, it sends a report back to the Puppet master, which helps the Puppet master in understanding where the current state of the system is, as defined in the catalog.

Puppet Use Cases

1. It can be used to manage infrastructure. It can be used to create new instances and configuring them
2. The main use case of puppet is for configuration management. It can be used to maintain certain configuration on the servers, update those configuration as needed and maintain these configuration histories over time.
3. It can be used for Application orchestration which enable you to create and manage multi-service and multi-node applications. Orchestration can be used to describe relationships between components of a distributed application.
4. Agents can be scheduled to check for configuration changes over time to automatically update the servers after making any changes to configuration.

Benefits

1. Well established support community through Puppet Labs. Puppet is considerably old system and hence has well established community for support. It also means, it has matured well and has been tested in various environments.
2. It is platform independent and can run on different Operating Systems. Although Puppet masters are rest Linux based Operating Systems, puppet agents are supported for almost all types of Operating Systems which makes it easier to include different kinds of servers into the same configuration groups.
3. Has a complete Web UI. Provides a simple web UI to manage puppet agents. This can be useful for people not familiar with specific Puppet functions.
4. Strong reporting capabilities.

Limitations:

1. You need to have puppet agent installed on each machine you want to configure. Whereas Ansible on the other hand doesn’t require any special software to be installed on the client nodes before being able to use it (apart from SSH server, Python etc. but with Linux, these usually ship along with the distro.
2. Uses a customized DSL Language to define configuration. This can make it difficult for people to get started.
3. The memory requirements for master node are comparatively high. On our local setup, we had to use minimum 4 GB RAM on vagrant VM.
4. Until Puppet 4.0, application orchestration was not supported.

Supported Operating Systems for Puppet:

Master Node: Red Hat Enterprise Linux, CentOS, Oracle Linux, Scientific Linux, SUSE Linux Enterprise Server, Ubuntu

Agent Nodes: Red Hat Enterprise Linux, CentOS, Oracle Linux, Scientific Linux, SUSE Linux Enterprise Server, Ubuntu, Solaris, Fedora, Debian, Windows, Mac OS X, AIX, Amazon Linux

Other Similar Products

Chef: Similar to Puppet, Chef operates as a master-client model, with a separate workstation needed to control the master. Unlike Puppet which uses customized Ruby, chef uses pure Ruby for most elements. Doesn't support push functionality. Code Driven – gives you more control over flexibility of configurations.

Ansible: Open Source tool used to deploy applications to remote nodes and provision servers in a repeatable way. No daemons need to be running on client nodes. It uses a push model setup providing a common framework to deploy applications. Has a rich collection of 'roles' on Ansible-galaxy which provides users with a set of pre-existing Ansible playbooks to set up different configurations on servers.

Fabric: Python based tool for streamlining SSH in application deployments. Can be used to run tasks across multiple remote systems. Helpful in system configuration, administration, and automatic deployment of applications. Especially useful for beginners who are just starting out in 'automatic deployment'.

Conclusion:

Should you use Puppet?
Depends on the size and type of infrastructure. If you have servers with different operating systems which need configuring, puppet provides support for a comprehensive list of operating systems.

Many of the advanced features are available only in Puppet Enterprise (which is not open source). If you are a small company which doesn't want to spend significantly on configuration management just yet, Ansible may be a better option.

References

Puppet Documentation: https://puppet.com/docs
Tutorial on Puppet: https://www.tutorialspoint.com/puppet
Similar Product comparison: https://www.intigua.com/blog/puppet-vs.-chef-vs.-ansible-vs.-saltstack
Similar Product comparison: http://blog.takipi.com/deployment-management-tools-chef-vs-puppet-vs-ansible-vs-saltstack-vs-fabric/