CS312 Homework #3

March 3, 2015

Instructions

1. (1pt) Describe why configuration management is important. What was one of the first configuration management system widely used?

   One of the first configuration management systems wildly used was CFEngine. Configuration management is important for many reasons: it allows for quick rebuilding of servers, it allows for guaranteeing the state of a server, and it allows the automation of configuration.

2. (1pt) Name four important problems that configuration management provides solutions for.

   A **Automating package installation**

   B **Standardizes policies and style**

   C **Provides a history of changes**

   D **Enables ops to define their infrastructure as code**

   E **Configure software**

   F **Start/stop/enable services**

   G **Ensures a state of a machine**

   H **Repeatable system builds**

   I **Assist in cluster orchestration**

3. (1pt) Name two primary differences between Puppet and Chef.

   A **Puppet uses a DSL, while Chef uses Ruby**

   B **Puppet is declarative, Chef is imperative**

4. (2pts) Given the following Chef code, explain in detail what its doing step by step. Is there a problem with this chef code, if so what is it?

```
service 'ntpd' do
  action [:enable, :start]
end

package 'ntpd' do
  package_name 'httpd'
  action :install
end
```

   This chef code enables and starts the ntpd service, and then installs the httpd package. There is a problem, which is that it tries to start the ntpd service but never installs the ntpd package.

5. (1pt) Explain the differences between push and pull models in configuration management. What are pros and cons of each?

   Pull models have nodes that run the client locally and request data from a server. Push models have nodes that are told when to run the client and with what data. Push models are simpler, but can be more difficult to scale. Pull models have a lot of overhead complexity that isn't necessary for a sufficiently small number of nodes, but handle large numbers of nodes much better.

6. (1pt) Name two configuration management tools that use push and two that use the pull method.

   Push: Ansible, Saltstack
   Pull: Chef, Puppet, CFEngine

7. (2pts) Give a brief description of each of the following components of chef. List any relationships between the components in their descriptions.

```
Resources
Providers
Nodes
Roles
Environments
```

Resources are a conceptualization of actions that should be taken. They are closely tied to providers, which tell the node how to take an action for any given resource. Nodes are the clients that run Chef. Each node has exactly one environment, and 0 or more roles. Roles are abstractions of node data that can apply to multiple nodes. Environments are abstractions of roles that generally apply to a large number of nodes at a time.

8. (1pt) The slides skipped over two types of node attributes. What are they? (Hint: Chef has very good documentation on the subject of attributes at `http://docs.chef.io`).

The slides skipped over `force_override` and `force_default`. These are used to reverse the attribute precedence order so that attributes defined in recipes take precedence over roles and environments

9. (1pt) Describe why templates are useful in the context of configuration management. Why should using files instead of templates be avoided?

Templates allow you to abstract configuration and other files in order to make them more broadly applicable. If using flat files, there will be a lot of repetition that can be avoided by the use of templates.

10. (1pt) Describe what Test Kitchen does and why its important.

Test Kitchen is a framework for running tests. It handles spinning up virtualized machines on a variety of providers (Openstack, AWS, Virtualbox, etc), running the chef cookbook you wish to test, and running the tests. It's important because it makes testing Chef cookbooks much simpler.

11. (3pts) Chef has a series of interesting podcasts on both Chef specifically and DevOps in general, located at `http://foodfightshow.org/blog/archives`. Listen to one entry that interests you. Write down the URL, a brief (1 paragraph) description of the podcast, and 3 things you learned.

    This answer is highly dependent on which podcast you listened to.

12. (5pts) Change the default recipe to pass all of the serverspec tests for the following cookbook: `https://github.com/osuosl/cs312-hw-cookbook`. Use Test Kitchen to converge, verify and test the recipe and run the tests. Once you have a recipe finished and passing all serverspec tests, tar up the entire cookbook into a file named `<onid>-hw3-cookbook.tar.gz` and attach it to your email along with the rest of your homework answers. Don't forget to utilize the Chef documentation page to help you out!

    Bonus: For 2 bonus points each,

    A. Use community cookbooks for Apache (httpd) and the iptables rule

    B. Ensure all ruby files pass Rubocop

    C. Ensure Foodcritic compliance

    One version of the solution has been pushed to `https://github.com/osuosl/cs312-hw-cookbook`, but it is not the only correct solution.