

Learn JAVA

TUESDAY, 4 APRIL 2017

DevOps Puppet Interview Questions And Answers.

DevOps Important Puppet Interview Questions And Answers.

1. How does merging work?

Answer: An external node Every node always gets a node object (which may be empty or may contain classes, parameters, and an environment) from the configured node_terminus. (This setting takes effect where the catalog is compiled; on the puppet master server when using an agent/master arrangement, and on the node, itself when using puppet apply. The default node terminus is plain, which returns an empty node object; the exec terminus calls an ENC script to determine what should go in the node object.) Every node may also get a node definition from the site manifest (usually called site.pp).

When compiling a node's catalog, Puppet will include all the following:

- Any classes specified in the node object it received from the node terminus
- Any classes or resources which are in the site manifest but outside any node definitions
- Any classes or resources in the most specific node definition in site.pp that matches the current node (if site.pp contains any node definitions)
- Note 1: If site.pp contains at least one node definition, it must have a node definition that matches the current node; compilation will fail if a match can't be found.
- Note 2: If the node name resembles a dot-separated fully qualified domain name, Puppet will make multiple attempts to match a node definition, removing the right-most part of the name each time. Thus, Puppet would first try agent1.example.com, then agent1.example, then agent1. This behavior isn't mimicked when calling an ENC, which is invoked only once with the agent's full node name.
- Note 3: If no matching node definition can be found with the node's name, Puppet will try one last time with a node name of default; most users include a node default {} statement in their site.pp file. This behavior isn't mimicked when calling an ENC.

2. Does PuppetDB support MySQL or Oracle?

Answer: No. It only supports PostgreSQL.

3. In relation to above question, why do you think Puppet does not support MySQL?

Answer: MySQL does not support recursive queries among other things.

4. What is storeconfigs?

Answer: Storeconfigs is a puppetmasterd option that stores the nodes actual configuration to a database. It does this by comparing the result of the last compilation against what is in the database, resource per resource, then parameter per parameter, and so on.

5. Why would I use storeconfigs?

Answer: The immediate use of storeconfigs is exported resources. Exported resources are resources which are prefixed by @@. Those resources are marked specially so that they can be collected on several other nodes. Some sample use cases are:

- Share/distribute public keys (ssh or openssl or other types)
- Build list of hosts running some services (for monitoring)
- Build configuration files which requires multiple hosts (for instance /etc/resolv.conf can be the concatenation of files exported by your dns cache hosts)

6. Explain what you mean by facter and some use case for same.

Answer: Sometime you need to write manifests on conditional expression based on agent specific data which is available through Facter. Facter provides information like Kernel version, Distribution release, IP Address, CPU info etc. You can define your own custom facts.

ABOUT ME



Madhu Sudhan Reddy Y.B

G+ Follow 40

[View my complete profile](#)

BLOG ARCHIVE

▼ 2017 (7)

► December (1)

► July (1)

▼ April (4)

[DevOps Docker and Network important interview ques...](#)

[DevOps Puppet Interview Questions And Answers.](#)

[DevOps RealTime interview Questions](#)

[DevOps Realtime Interview Questions](#)

► March (1)

► 2016 (6)

► 2015 (7)

Facter can be used independently from Puppet to gather information about a system. Whether it's parsing the `/proc/xen` directory on Linux or running `prttdiag` command on Solaris, the tool does a great job abstracting the specific operating system commands used to determine the collection of facts. When used in conjunction with Puppet, facts gather through the system allows the puppet master to make intelligent decisions during manifest compilation. Within your puppet manifest, you can reference any key value pairs provided by facter by prefixing the hash key with "\$"

If the default set of facts are not sufficient, there are two ways to extend Facter to provide additional fact. One way is to use Ruby, the other way is by using environment variables prefixed with `FACTER_`. This allows one to set per node data and utilize it in our manifests.

7. What is Manifests?

Answer: Manifests, in Puppet, are the files in which the client configuration is specified.

8. What is MCollective?

Answer: MCollective is a powerful orchestration framework. Run actions on thousands of servers simultaneously, using existing plugins or writing your own.

9. What are classes?

Answer: Classes are named blocks of Puppet code that are stored in modules for later use and are not applied until they are invoked by name. They can be added to a node's catalog by either declaring them in your manifests or assigning them from an ENC.

Classes generally configure large or medium-sized chunks of functionality, such as all of the packages, config files, and services needed to run an application.

10. How do you test your manifest files?

Answer: Explain how you will first run syntax checks with puppet parser validate command. If you are using VIM, you can use plugins like Syntastic to verify code (or else) use a full-fledged IDE like Geppetto. Also use puppet lint to verify in addition to puppet parser.

You can add rspec/cucumber tests to your application. (BE CAREFUL: RUBY skills are required) and use Cucumber in standalone mode to test your manifests.

11. What are some messaging systems I can use with MCollective?

Answer: RabbitMQ and ActiveMQ are common options.

12. Have you worked with MCollective? Why would you use it over plain vanilla Puppet?

Answer: The Marionette Collective, also known as MCollective, is a framework for building server orchestration or parallel job-execution systems. Most users programmatically execute administrative tasks on clusters of servers.

MCollective has some unique strengths for working with large numbers of servers:

- Instead of relying on a static list of hosts to command, it uses metadata-based discovery and filtering. It can use a rich data source like PuppetDB, or can perform real-time discovery across the network.

Instead of directly connecting to each host (which can be resource-intensive and slow), it uses publish/subscribe middleware to communicate in parallel with many hosts at once.

13. Write a simple ENC script

Answer:

Create one YAML file in `/tmp/node-definitions` with the classes listed for node. The name of the file is `<FQDN>.yaml`. The parameter is passed by Puppet Master automatically. To configure Puppet master, edit `puppet.conf` and add following lines:

14. What are the limitations of using ENC?

Answer: The YAML returned by an ENC isn't an exact equivalent of a node definition in `site.pp` — it can't declare individual resources, declare relationships, or do conditional logic. The only things an ENC can do are declare classes, assign top-scope variables, and set an environment. This means an ENC is most effective if you've done a good job of separating your configurations out into classes and modules.

Unlike regular node definitions, where a node may match a less specific definition if an exactly matching one isn't found (depending on the puppet

master's strict_hostname_checking setting), an ENC is called only once, with the node's full name.

15. What is HIERA?

Answer: It's key/value lookup tool.

16. How to configure HIERA?

Answer: (i) Create a Hiera.yaml file (ii) Create a hierarchy. Each Element in a hierarchy is a data source.

17. What data source types are supported in HIERA?

Answer: JSON and YAML

18. What is the LDAP Node Classifier?

Answer: LDAP can be used to store information about nodes and servers. The LDAP Node Classifier is used query LDAP for node information instead of an ENC.

19. What is the use of etckeeper-commit-post and etckeeper-commit-pre on Puppet Agent?

Answer: etckeeper-commit-post: In this configuration file, you can define command and scripts which executes after pushing configuration on Agent
Etckeeper-commit-pre: In this configuration file you can define command and scripts which executes before pushing configuration on Agent

20. What is Puppet Kick?

Answer: By default, Puppet Agent request to Puppet Master after a periodic time which known as "runinterval". Puppet Kick is a utility which allows you to trigger Puppet Agent from Puppet Master.

21. Explain differences in class definition vs declaration.

Answer: Defining a class makes it available for later use. It doesn't yet add any resources to the catalog; to do that, you must declare it or assign it from an ENC.

22. Explain what you mean by ordering and relationships?.

Answer: By default, Puppet applies resources in the order they're declared in their manifest. However, if a group of resources must always be managed in a specific order, you should explicitly declare such relationships with relationship metaparameters, chaining arrows, and the require function. Puppet uses four metaparameters to establish relationships, and you can set each of them as an attribute in any resource. The value of any relationship metaparameter should be a resource reference (or array of references) pointing to one or more target resources. □ before - Applies a resource before the target resource. □ require - Applies a resource after the target resource. □ notify - Applies a resource before the target resource. The target resource refreshes if the notifying resource changes. □ subscribe - Applies a resource after the target resource. The subscribing resource refreshes if the target resource changes.

If two resources need to happen in order, you can either put a before attribute in the prior one or a require attribute in the subsequent one; either approach creates the same relationship. The same is true of notify and subscribe.

23. What design patterns have you used in your Puppet Code?

Answer: Tip to answer: Be very clear on this as this tests your development knowledge instead of sysadmin skills. At very least, you will be quizzed deeply on role/profile patterns – this is the best and most complex approach to maintaining code. Mention Anchor pattern and explain why you are using it. You should only be using this pattern prior to Puppet 3.4 or PE 3.2.

24. Explain some best practices for Puppet

Answer: Some best practices are listed below: 1. Use Modules and Role/Profile pattern as much as possible. 2. Keep all code in version control with published modules/manifests being sourced from officially tested/released branches. 3. Make use of environments - Each environment is tied to a specific branch in the version control system. 4. Use Dry runs – use puppet agent -verbose -noop -test. 5. Manage puppet module dependencies using librarian puppet. Version control and manage your Puppetfile 6. Keep data and code separate. Specifically keep sensitive data inside Hiera YAML files (which are also version controlled). 7. Trick to doing the above is to use JSON files for confidential data and YAML for non-confidential data and ensure physical security to the JSON Folder where no one except root/puppet user can access.

25. What is r10k?

Answer: R10k provides a general-purpose toolset for deploying Puppet environments and modules. It implements the Puppetfile format and provides a native implementation of Puppet dynamic environments. R10k has two primary roles: installing Puppet modules using a standalone Puppetfile, and managing Git and SVN based dynamic environments

26. What would be a common workflow for using r10K?

Answer: Much of the common workflow depends on the standard gitflow to creating feature branches and making all changes in feature branches, deploying the environment in test mode, and on verification, merging to master branch before cleaning up feature branch. If you are not using the standard git flow, adopt accordingly.

Posted by [Madhu Sudhan Reddy Y.B](#) at [19:53](#)



Labels: [DevOps](#), [Puppet](#)

2 comments:



Mr Tic 22 November 2017 at 17:20

Tks very much for your post.

Avoid surprises — interviews need preparation. Some questions come up time and time again — usually about you, your experience and the job itself. We've gathered together the most common questions so you can get your preparation off to a flying start.

You also find all interview questions at link at the end of this post.

Source: [Download Ebook: Ultimate Guide To Job Interview Questions Answers:](#)

Best rgs

[Reply](#)



soumya Teja 15 February 2018 at 02:48

You have shared wonderful information with us, Thanks for sharing keep share. Check it once through [Devops Online Training](#)

[Reply](#)

Enter your comment...



Comment as: [Unknown \(Goo](#) ▼

[Sign out](#)

[Publish](#)

[Preview](#)

☐ [Notify me](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)