***Dev-Ops: Interview Preparation***

1. How does HTTP work?
2. Explain your understanding and expertise on both the software development side and the technical operations side of an organization you have worked for the past?
3. Discuss your experience building bridges between IT Ops, QA and development?
4. What types of testing are needed?
5. Give me an example of how you would handle projects?
6. What's your career objective in your role as a DevOps engineer?
7. How would you make software deployable?
8. What is the most important thing DevOps helps do?
9. Which scripting languages do you think are most important for a DevOps engineer?
10. How do you expert you would be required to multitask as a DevOps professional?
11. What testing is necessary to ensure that a new service is ready for production?
12. What's a PTR in DNS?
13. Describe two-factor authentication?
14. Tell us about CI tools that you are familiar with?
15. What are advantages of NoSQL database over RDBMS?
16. What is an MX record in DNS?
17. What is the difference between RAID 0 and RAID 1?
18. How would you prepare for a migration?
19. What's your systems background?
20. What DevOps tools have you worked with?
21. How much have you interacted with cloud based software development?
22. What other tools are you familiar with that might help you in this role?
23. Are you familiar with just Linux or have you worked with Windows environments as well?
24. How can you reduce load time of a dynamic website?
25. Describe your experience implementing continuous deployment?
26. How would you ensure traceability?
27. What was your greatest achievement on a recent project?
28. What problems did you face and how did you solve them in a way that met the team's goals?

**PUPPET**

1. What is Puppet?
2. What is Manifests?
3. What is Module? and How it is different from Manifest?
4. Command to check requests of Certificates?
5. Command to sign requested Certificates?
6. Where Puppet Master Stores Certificates?
7. What is Facter?
8. What is the use of etckeeper-commit-post and etckeeper-commit-pre on Puppet Agent?
9. What is Puppet Kick?
10. What is MCollective?
11. What's special about Puppet's model driven design?
12. Why does Puppet have its own language?
13. Can Puppet manage workstations?
14. Does Puppet run on Windows?
15. What size organizations should use Puppet?
16. My servers are all unique; can Puppet still help?
17. Who is puppet Labs?
18. How should I update puppet and Facter?
19. What characters are permitted in a class name? In a module name? In other identifiers?
20. How do I document my manifests?
21. How do I manage passwords on Red Hat enterprise Linux, CentOS, and Fedora Core?
22. How do all of these variables, like $operatingsystem, get set?
23. Can I access environment variables with Facter?
24. Why shouldn't I use auto-sign for all my clients?
25. What happens if I am on puppet 2.6x or earlier?
26. Does this change affect all the components of puppets?
27. What does it mean if I or my company have or want to contribute code to puppet?
28. What if I haven't signed a CLA?
29. Does signing a CLA change who owns puppet?
30. Which versions of Ruby does puppet support?
31. What is Virtual Box? What is Vagrant?

**Docker**

1. What is Docker?
2. What is the advantage of Docker over hypervisors?
3. What is Docker image?
4. What is Docker container?
5. Difference between Docker image and container?

**AWS**

1. Explain what is AWS?
2. Mention what are the key components of AWS?
3. Explain what is S3?
4. Explain what is AMI?
5. Mention what is the relation between an instance and AMI?
6. What does an AMI include?
7. How can you send request to Amazon S3?
8. Mention what is the difference between Amazon S3 and EC2?
9. How many bucket can you create in AWS by default?
10. Explain can you vertically scale an Amazon instance? How?
11. Explain what is T2 instance?
12. In VPC with private and public subnets, database servers should ideally be launched into which subnet?
13. Mention what are the security best practices for Amazon EC2?
14. Explain how the buffer is used in Amazon web services?
15. While connecting to your instance what are the possible connections issues one might face?

**Chef**

1. What is resource?
2. What is recipe?
3. What happens when you don't specify a resource's action?
4. Are these two recipes same?

package 'httpd'

service 'httpd' do

action [ :enable, :start ]

end

&&

service 'httpd' do

action [ :enable, :start ]

end

package 'httpd'

1. Are these two recipes same?

package 'httpd'

service 'httpd' do

action [ :enable, :start ]

end

&&

package 'httpd'

service 'httpd' do

action [ :start, :enable ]

end

1. Are these two recipes same?

file '/etc/motd/' do

owner 'root'

group 'root'

mode '0755'

action :create

end

&&

file '/etc/motd/' do

action :create

mode '0755'

owner 'root'

group 'root'

end

1. Write a service resource that stops and then disables the httpd service from starting when the system boots?
2. How does a cookbook differ from a recipe?
3. How does chef-apply differ from chef-client?
4. What's a Run-List?
5. What are the two ways to set up a chef server?
6. What's the role of the starter Kit?
7. Where can you get reusable cookbooks that are written and maintained by chef community?
8. What's the command that enables you to interact with the chef server?
9. What is a node?
10. What information do you need to in order to bootstrap?
11. What happens during the bootstrap process?
12. Which of the following lets you verify that your node is successfully bootstrapped?
13. What is the command you use to upload a cookbook to the chef server?
14. How do you apply an updated cookbook to your node?
15. What would you set your cookbook's version to once it's ready to use in production?
16. What is the latest version of the haproxy community cookbook?
17. Create a second node and apply the awesome\_customers cookbook to it. How long does it take?
18. What's the value of local development using test kitchen?