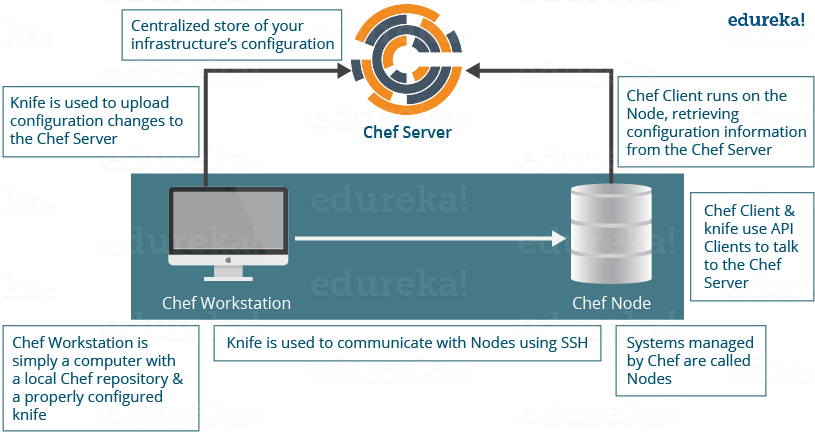
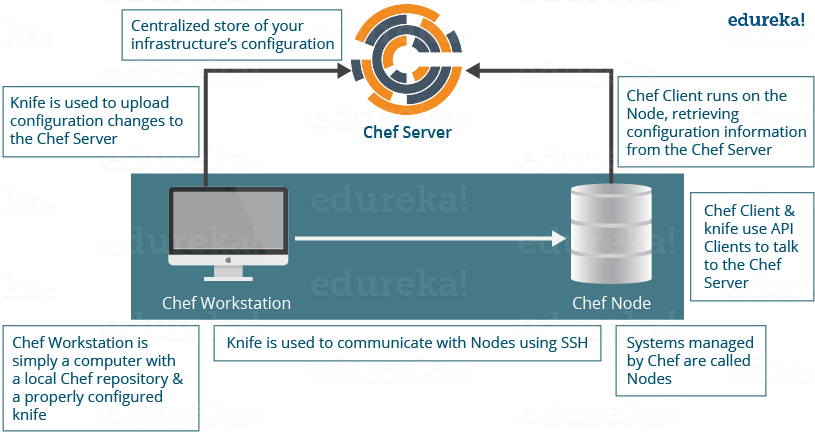
**Q1. What is Chef?**

Begin this answer by defining Chef.

It is a powerful automation platform that provides a way to transforms infrastructure into code. Chef is a tool for which you write scripts that are used to automate processes. What processes? Pretty much anything related to IT.

Now you can explain the architecture of Chef, it consists of:

* **Chef Server:**The Chef Server is the central store of your infrastructure’s configuration data. The Chef Server stores the data necessary to configure your nodes and provides search, a powerful tool that allows you to dynamically drive node configuration based on data.
* **Chef Node:** A Node is any host that is configured using Chef-client. Chef-client runs on your nodes, contacting the Chef Server for the information necessary to configure the node. Since a Node is a machine that runs the Chef-client software, nodes are sometimes referred to as “clients”.
* **Chef Workstation:** A Chef Workstation is the host you use to modify your cookbooks and other configuration data. All the configurations are first tested in the Chef Workstation and then it is forwarded to the Chef Server.



**Q1. What is a Resource in Chef?**

My suggestion is to first define Resource.

A Resource represents a piece of infrastructure and its desired state, such as a package that should be installed, a service that should be running, or a file that should be generated. A block of Resource can be considered as a Recipe.

Now you should explain about the functions of Resource for that include the following points:

* Describes the desired state for a configuration item.
* Declares the steps needed to bring that item to the desired state.
* Specifies a resource type such as package, template, or service.
* Lists additional details (also known as resource properties), as necessary.
* Are grouped into recipes, which describe working configurations.

*Remember, you have mentioned the word Recipe in your previous answer, so the next question in this Chef interview questions blog has to be related to Recipe.*

**Q2. What is a Recipe in Chef?**

Here also I will suggest you to use the above-mentioned flow, first define Recipe.

A Recipe is a collection of Resources that describes a particular configuration or policy. A Recipe describes everything that is required to configure part of a system.

Now after the definition I will explain the functions of Recipes by including the following points:

* Install and configure software components.
* Manage files.
* Deploy applications.
* Execute other Recipes.

**Q3. What is a Node in Chef?**

This will be probably the easiest question you can encounter answer this by saying:

A Node represents a server and is typically a virtual machine, container instance, or physical server – basically any compute resource in your infrastructure that is managed by Chef.

**Q4.** **How does a Cookbook differ from a Recipe in Chef?**

The answer to this is pretty direct My suggestion is to simply tell:

A Recipe is a collection of Resources, and primarily configures a software package or some piece of infrastructure. A Cookbook groups together Recipes and other information in a way that is more manageable than having just Recipes alone.

*Now the following set of Chef interview questions are to test your experience with Chef:*

**Q5.** **What happens when you don’t specify a Resource’s action in Chef?**

My suggestion is to first give a direct answer.

When you don’t specify a resource’s action, Chef applies the default action.

Now explain this with an example, the below resource:

|  |  |
| --- | --- |
| 1  2  3 | file 'C:\Users\Administrator\chef-repo\settings.ini' do   content 'greeting=hello world'    end |

is same as the below resource:

|  |  |
| --- | --- |
| 1  2  3  4 | file 'C:\Users\Administrator\chef-repo\settings.ini' do  action :create  content 'greeting=hello world'  end |

because: create is the file Resource’s default action.

**Q6. Are these two Chef recipes the same?**

|  |  |
| --- | --- |
| 1  2  3  4 | package 'httpd'   service 'httpd' do   action [:enable, :start]   end |

**&&**

|  |  |
| --- | --- |
| 1  2  3  4 | service 'httpd' do  action [:enable, :start]  end  package 'httpd' |

No, they are not. Remember that Chef applies resources in the order they appear. So the first Recipe ensures that the httpd package is installed and then configures the service. The second Recipe configures the service and then ensures the package is installed.

**Q7. Write a service Resource that stops and then disables the httpd service from starting when the system boots in Chef.**

Use the below Resource to stop and disable the httpd service from starting when system boots.

|  |  |
| --- | --- |
| 1  2  3 | service 'httpd' do  action [:stop, :disable]    end |

**Q8.** **How does Chef-apply differ from Chef-client?**

I suggest you to follow the below mentioned flow to answer this question:

Chef-apply is an executable program that runs a single Recipe from the command line. It is a part of the Chef development kit and a great way to explore resources.

Syntax for Chef-apply is:

|  |  |
| --- | --- |
| 1 | chef-apply name\_of\_recipe.rb |

Chef-client applies a Cookbook. It is used for production purposes where you typically run Chef-client to apply one or more cookbooks.

**Q9. What is run-list in Chef?**

run-list lets you specify which Recipes to run, and the order in which to run them. The run-list is important when you have multiple Cookbooks, and the order in which they run matters.

Depending on the discussion if you think more explanation is required just mention the below points

A run-list is:

* An ordered list of roles and/or recipes that are run in the exact order defined in the run-list; if a recipe appears more than once in the run-list, the chef-client will not run it twice.
* Always specific to the node on which it runs; nodes may have a run-list that is identical to the run-list used by other nodes.
* Stored as part of the node object on the Chef server.
* Maintained using knife, and then uploaded from the workstation to the Chef server, or is maintained using the Chef management console.

**Q10.** **What information do you need in order to bootstrap in Chef?**

Just mention the information you need in order to bootstrap:

* Your node’s host name or public IP address.
* A user name and password you can log on to your node with.
* Alternatively, you can use key-based authentication instead of providing a user name and password.

**Q11.** **How do you apply an updated Cookbook to your node in Chef?**

There are three ways to apply an updated Cookbook to a node you can mention all or any one, I will suggest you to mention all three:

* Run knife ssh from your workstation.
* SSH directly into your server and run chef-client.
* You can also run chef-client as a daemon, or service, to check in with the Chef server on a regular interval, say every 15 or 30 minutes.

**Q12.** **What is the role of Starter Kit in Chef?**

Begin this answer by mentioning the functions of Starter Kit.

Starter Kit will create the necessary configuration files like chef directory, knife.rb, the ORGANIZATION-validator.pem, and USER.pem files etc. with the correct information that is required to interact with the Chef server.

Now tell how to use Starter Kit, you can simply download the starter kit and then move it to the desired location on your workstation.

**Q13.** **What is the command you use to upload a cookbook to the Chef server?**

You can directly mention the command to upload a cookbook to the Chef server **“knife cookbook upload”**.

**Q14. What would you set your cookbook’s version to once it is ready to use in production?**

According to Semantic Versioning, you should set your cookbook’s version number to 1.0.0 once it is ready to use in production.

**Q15.** **What is the value of local development using Test Kitchen in Chef?**

I will mention the below points, this will give the interviewer a clear picture of your understanding of Test Kitchen.

* Test Kitchen enables you to use a variety of virtualization providers that create virtual machine or container instances locally on your workstation or in the cloud.
* It enables you to run your cookbooks on servers that resemble those that you use in production.
* It speeds up the development cycle by automatically provisioning and tearing down temporary instances, resolving cookbook dependencies, and applying your cookbooks to your instances.

**Q16. Where can you get reusable cookbooks that are written and maintained by the Chef community?**

You can directly answer this question by saying reusable Cookbooks are present at Chef Supermarket,[***https://supermarket.chef.io****.*](https://supermarket.chef.io./)

*-----------------------------------------------------------------------------------*

##################### 2 ##########################

### ****Q.How does HTTP work?****

The HTTP protocol works in a client and server model like most other protocols. A web browser using which a request is initiated is called as a client and a web server software which responds to that request is called a server. World Wide Web Consortium and the Internet Engineering Task Force are two important spokes in the standardization of the HTTP protocol. HTTP allows improvement of its request and response with the help of intermediates, for example a gateway, a proxy, or a tunnel. The resources that can be requested using the HTTP protocol, are made available using a certain type of URI (Uniform Resource Identifier) called a URL (Uniform Resource Locator). TCP (Transmission Control Protocol) is used to establish a connection to the application layer port 80 used by HTTP.

### ****Explain your understanding and expertise on both the software development side and the technical operations side of an organization you’ve worked for in the past.****

DevOps engineers almost always work in a 24/7 business critical online environment. I was adaptable to on-call duties and able to take up real-time, live-system responsibility. I successfully automated processes to support continuous software deployments. I have experience with public/private clouds, tools like Chef or Puppet, scripting and automation with tools like Python and PHP, and a background in Agile.

### ****Discuss your experience building bridges between IT Ops, QA and development.****

DevOps is all about effective communication and collaboration. I’ve been able to deal with production issues from the development and operations sides, effectively straddling the two worlds. I’m less interested in finding blame or playing the hero than I am with ensuring that all of the moving parts come together.

### ****What types of testing are needed?****

Software teams will often look for the “fair weather” path to system completion; that is, they start from an assumption that software will usually work and only occasionally fail. I believe to practice defensive programming in a pragmatic way, which often means assuming that the code will fail and planning for those failures. I try to incorporate unit test strategy, use of test harnesses, early load testing; network simulation, A/B and multi-variate testing  etc.

**Give me an example of how you would handle projects?**

As a professional with managerial responsibilities, I would demonstrate a clear understanding of DevOps project management tactics and also work with teams to set objectives, streamline workflow, maintain scope, research and introduce new tools or frameworks, translate requirements into workflow and follow up. I would resort to CI, release management and other tools to keep interdisciplinary projects on track.

### ****What’s your career objective in your role as a DevOps engineer?****

My passion is breaking down the barriers and building and improving processes, so that the engineering and operations teams work better and smarter. That’s why I love DevOps. It’s an opportunity to be involved in the entire delivery system from start to finish.

### ****How would you make software deployable?****

The ability to script the installation and reconfiguration of software systems is essential towards controlled and automated change. Although there is an increasing trend for new software to enable this, older systems and products suffer from the assumption that changes would be infrequent and minor, and so make automated changes difficult. As a professional who appreciates the need to expose configuration and settings in a manner accessible to automation, I will work with concepts like Inversion of Control (IoC) and Dependency Injection, scripted installation, test harnesses, separation of concerns, command-line tools, and infrastructure as code.

### ****What is the one most important thing DevOps helps do?****

The most important thing DevOps helps do is to get the changes into production as quickly as possible while minimizing risks in software quality assurance and compliance. That is the primary objective of DevOps. However, there are many other positive side-effects to DevOps. For example, clear communication and better working relationships between teams which creates a less stressful working environment.

### ****Which scripting languages do you think are most important for a DevOps engineer?****

As far as scripting languages go, the simpler the better. In fact, the language itself isn’t as important as understanding design patterns and development paradigms such as procedural, object-oriented, or functional programming.

### ****How do you expect you would be required to multitask as a DevOps professional?****

I believe I’ll be expected to:

1. Focus attention on bridging communication gaps between Development and Operations teams.
2. Understand system design from an architect’s perspective, software development from a developer’s perspective, operations and infrastructure from the perspective of a seasoned Systems Administrator.
3. Execute – to be able to actually do what needs to be done.

### ****What testing is necessary to ensure that a new service is ready for production?****

DevOps is all about continuous testing throughout the process, starting with development through to production. Everyone shares the testing responsibility. This ensures that developers are delivering code that doesn’t have any errors and is of high quality, and it also helps everyone leverage their time most effectively.

### ****What’s a PTR in DNS?****

Pointer records are used to map a network interface (IP) to a host name. These are primarily used for reverse DNS. Reverse DNS is setup very similar to how normal (forward) DNS is setup.  When you delegate the DNS forward, the owner of the domain tells the registrar to let your domain use specific name servers.

### ****Describe two-factor authentication?****

Two-factor authentication is a security process in which the user provides two means of identification from separate categories of credentials; one is typically a physical token, such as a card, and the other is typically something memorized, such as a security code.

### ****Tell us about the CI tools that you are familiar with?****

The premise of CI is to get feedback as early as possible because the earlier you get feedback, the less things cost to fix. Popular open source tools include Hudson, Jenkins, CruiseControl and CruiseControl.NET. Commercial tools include ThoughtWorks’ Go, Urbancode’s Anthill Pro, Jetbrains’ Team City and Microsoft’s Team Foundation Server.

### ****What are the advantages of NoSQL database over RDBMS?****

The advantages are:

1. Less need for ETL
2. Support for unstructured text
3. Ability to handle change over  time
4. Breadth of functionality
5. Ability to scale horizontally
6. Support for multiple  data structures
7. Choice of vendors

### ****What is an MX record in DNS?****

MX records are mail exchange records used for determining the priority of email servers for a domain. The lowest priority email server is the first destination for email. If the lowest priority email server is unavailable, mail will be sent to the higher priority email servers.

### ****What is the difference between RAID 0 and RAID 1?****

RAID 1 offers redundancy through mirroring, i.e., data is written identically to two drives. RAID 0 offers no redundancy and instead uses striping, i.e., data is split across all the drives. This means RAID 0 offers no fault tolerance; if any of the constituent drives fails, the RAID unit fails.

### ****How would you prepare for a migration?****

Tips to answer: This question evaluates your experience of real projects with all the awkwardness and complexity they bring. Include terms like cut-over, dress rehearsals, roll-back and roll-forward, DNS solutions, feature toggles, branch by abstraction, and automation in your answer. Developing greenfield systems with little or no existing technology in place is always easier than having to deal with legacy components and configuration. As a candidate if you appreciate that any interesting software system will in effect be under constant migration, you will appear suitable for the role.

### ****What’s your systems background?****

Tips to answer: Some DevOps jobs require extensive systems knowledge, including server clustering and highly concurrent systems. As a DevOps engineer, you need to analyze system capabilities and implement upgrades for efficiency, scalability and stability, or resilience. It is recommended that you have a solid knowledge of OSes and supporting technologies, like network security, virtual private networks and proxy server configuration.

DevOps relies on virtualization for rapid workload provisioning and allocating compute resources to new VMs to support the next rollout, so it is useful to have in-depth knowledge around popular hypervisors. This should ideally include backup, migration and lifecycle management tactics to protect, optimize and eventually recover computing resources. Some environments may emphasize microservices software development tailored for virtual containers. Operations expertise must include extensive knowledge of systems management tools like Microsoft System Center, Puppet, Nagios and Chef. DevOps jobs with an emphasis on operations require detailed problem-solving, troubleshooting and analytical skills.

### ****What DevOp tools have you worked with?****

Tips to answer: Software configuration management and build/release (version control) tools, including Apache Subversion, Mercurial, Fossil and others, help document change requests. Developers can more easily follow the company’s best practices and policies while software changes.

Continuous integration (CI) tools such as Rational Build Forge, Jenkins and Semaphore merge all developer copies of the working code into a central version. These tools are important for larger groups where teams of developers work on the same codebase simultaneously. QA experts use code analyzers to test software for bugs, security and performance. If you’ve used HP’s Fortify Static Code Analyzer, talk about how it identified security vulnerabilities in coding languages. Also speak about tools like GrammaTech’s CodeSonar that you used to identify memory leaks, buffer underruns and other defects for C/C++ and Java code. It is essential that you have adequate command of the principal languages like Ruby, C#, .NET, Perl, Python, Java, PHP, Windows PowerShell, and are comfortable with the associated OS environments Windows, Linux and Unix.

### ****How much have you interacted with cloud based software development?****

Tips to answer: Share your knowledge around use of cloud platforms, provisioning new instances, coding new software iterations with the cloud provider’s APIs or software development kits, configuring clusters to scale computing capacity, managing workload lifecycles and so on. This is the perfect opportunity to discuss container-based cloud instances as an alternative to conventional VMs. Event-based cloud computing, such as AWS Lambda offers another approach to software development, a boon for experienced DevOps candidates. In your interview, mention experience handling big data, which uses highly scalable cloud infrastructures to tackle complex computing tasks.

### ****What other tools are you familiar with that might help you in this role?****

Tips to answer: DevOps is so diverse and inclusive that it rarely ends with coding, testing and systems. A DevOps project might rely on database platforms like SQL or NoSQL, data structure servers like Redis, or configuration and management issue tracking systems like Redmine. Web applications are popular for modern enterprises, making a background with Web servers, like Microsoft Internet Information Services, Apache Tomcat or other Web servers, beneficial. Make sure to bring across that you are familiar with Agile application lifecycle management techniques and tools.

### ****Are you familiar with just Linux or have you worked with Windows environments as well?****

Tips to answer: Demonstrate as much as you can, a clear understanding of both the environments including the key tools.

### ****How can you reduce load time of a dynamic website?****

Tips to answer: Talk about Webpage optimization, cached web pages, quality web hosting, compressed text files, Apache fine tuning.

### ****Describe your experience implementing continuous deployment?****

Tips to answer: Answer with a comprehensive list of all the tools that you used. Include inferences of the challenges you faced and how you tackled them.

### ****How would you ensure traceability?****

Tips to answer: This question probes your attitude to metrics, logging, transaction journeys, and reporting. You should be able to identify that metric, monitoring and logging needs to be a core part of the software system, and that without them, the software is essentially not going to be able to appear maintained and diagnosed. Include words like SysLog, Splunk, error tracking, Nagios, SCOM, Avicode in your answer.

### ****What was your greatest achievement on a recent project?****

Tips to answer: Make sure you demonstrate your perfect understanding of both development and operations. Do not let your answer lean towards one particular skillset ignoring the other. Even if you have worked in an environment wherein you had to work more with one skillset, assure the interviewer that you are agile according to the needs of your organization.

### ****What problems did you face and how did you solve them in a way that met the team’s goals?****

Tips to answer: This question aims to find out how much you can handle stress and non-conformity at work. Talk about your leadership skills to handle and motivate the team to solve problems together. Talk about CI, release management and other tools to keep interdisciplinary projects on track.

### ****Are you more Dev or Ops?****

Tips to answer: This is probably the trickiest question that you might face in the interview. Emphasize the fact that this depends a lot on the job, the company you are working for and the skills of people involved. You really have to be able to alternate between both sides of the fence at any given time. Talk about your experience and demonstrate how you are agile with both.

### ****What special training or education did it require for you to become a DevOps engineer?****

Tips to answer: DevOps is more of a mind-set or philosophy rather than a skill-set. The typical technical skills associated with DevOps Engineers today is Linux systems administration, scripting, and experience with one of the many continuous integration or configuration management tools like Jenkins and Chef. What it all boils down to is that whatever skill-sets you have, while important, are not as important as having the ability to learn new skills quickly to meet the needs. It’s all about pattern recognition, and having the ability to merge your experiences with current requirements. Proficiency in Windows and Linux systems administration, script development, an understanding of structured programming and object-oriented design, and experience creating and consuming RESTful APIs would take one a long way.