Chef Devops

**1. What is Chef?**  
**Answer:** It is a powerful automation platform that provides a way to transforms infrastructure into code. A 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 a 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.

**2. What is a Resource in Chef?**  
**Answer:** 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.

**3. What is a Recipe in Chef?**  
**Answer:** 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.

**4. What is a Node in Chef?**  
**Answer:** A Node represents a server and is typically a virtual machine, container instance, or physical server – basically, any computer resource in your infrastructure that is managed by Chef.

**5. How does a Cookbook differ from a Recipe in Chef?**  
**Answer:** 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.

**6. What is the architecture of Chef?**  
**Answer:** Chef is composed of many components like Chef Server, Client, etc. Some of the main components in Chef are as follows:

**Client:** These are the nodes or individual users that communicate with Chef server.

**Chef Manage:** This is the web console that is used for interacting with Chef Server.

**Load Balancer:** All the Chef server API requests are routed through Load Balancer. It is implemented in Nginx.

**Bookshelf:** This is the component that stores cookbooks. All the cookbooks are stored in a repository. It is separate storage from the Chef server.

**PostgreSQL:** This is the data repository for Chef server.

**Chef Server:** This is the hub for configuration data. All the cookbooks and policies are stored in it. It can scale to the size of any enterprise.

**7. What are the main benefits of Chef?**  
**Answer:** a Chef is an automation tool for keeping the infrastructure as code. It has many benefits.

**Some of these are as follows:**

**Cloud Deployment:** We can use Chef to perform automated deployment in Cloud environment.

**Multi-cloud support:** With Chef we can even use multiple cloud providers for our infrastructure.

**Hybrid Deployment:** Chef supports both Cloud-based as well as data center-based infrastructure.

**High Availability:** With Chef automation, we can create a high availability environment. In a state of hardware malfunction, Chef can control or begin supplementary servers in an automated process to sustain an extremely accessible atmosphere.

**8. How does Chef-apply differ from Chef-client?**  
**Answer:** 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.

**The syntax for Chef-apply is:**

1chef-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.

**9. What is run-list in Chef?**  
**Answer:** 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 a knife, and then uploaded from the workstation to the Chef server, or is maintained using the Chef management console.

**10. What information do you need in order to bootstrap in Chef?**  
**Answer:** Just mention the information you need in order to bootstrap:

* Your node’s hostname 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.

**11. How do you apply an updated Cookbook to your node in Chef?**  
**Answer:** There are three ways to apply an updated Cookbook to a node you can mention all or anyone, I will suggest you 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.

**12. What is the role of Starter Kit in Chef?**  
**Answer:** 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.

**13. What is the command you use to upload a cookbook to the Chef server?**  
**Answer:** You can directly mention the command to upload a cookbook to the Chef server “knife cookbook upload”.

**14. What would you set your cookbook’s version to once it is ready to use in production?**  
**Answer:** According to Semantic Versioning, you should set your cookbook’s version number to 1.0.0 once it is ready to use in production.

**15. What is the value of local development using Test Kitchen in Chef?**  
**Answer:** 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.

**16. How Does Http Work?**  
**Answer:** 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.

**17. 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? Answer:** 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.

**18. Discuss Your Experience Building Bridges Between It Ops, Qa, And Development?**  
**Answer:** [DevOps](https://devops.com/" \t "_blank) 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.

**19. What Types Of Testing Are Needed?**  
**Answer:** 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 multivariate testing, etc.

**20. Give Me An Example Of How You Would Handle Projects?**  
**Answer:** 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 the workflow and follow up. I would resort to CI, release management and other tools to keep interdisciplinary projects on track.

**21. What Is The One Most Important Thing Devops Helps Do?**  
**Answer:** 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, clearer communication and better working relationships between teams which creates a less stressful working environment.

**22. Which Scripting Languages Do You Think Are Most Important For A Devops Engineer?**  
**Answer:** 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.

**23. How Do You Expect You Would Be Required To Multitask As A Devops Professional?**  
**Answer:** I believe I’ll be expected to:

Focus attention on bridging communication gaps between Development and Operations teams.

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.

Execute – to be able to actually do what needs to be done.

**24. What Testing Is Necessary To Ensure That A New Service Is Ready For Production?**  
**Answer:** 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. [Devops Training](https://svrtechnologies.com/devops-training" \t "_blank) Free Demo

**25. Tell Us About The Ci Tools That You Are Familiar With?**  
**Answer:** The premise of CI is to get feedback as early as possible because the earlier you get feedback, the fewer 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.

**26. What is an MX record in DNS?**  
**Answer:** 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.

**27. What Is The Difference Between Raid 0 And Raid 1?**  
**Answer:** 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.

**28. What’s a PTR in DNS?**  
**Answer:** Pointer records are used to map a network interface (IP) to a hostname. These are primarily used for reverse DNS. Reverse DNS is set up 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.

**29. What are the advantages of NoSQL database over RDBMS?**  
**Answer:** 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

**30. What is the most important thing DevOps helps us achieve?**  
**Answer:** According to me, the most important thing that DevOps helps us achieve is to get the changes into production as quickly as possible while minimizing risks in software quality assurance and compliance. This is the primary objective of DevOps. Learn more in this DevOps tutorial blog.

However, you can add many other positive effects of DevOps. For example, clearer communication and better working relationships between teams i.e. both the Ops team and Dev team collaborate together to deliver good quality software which in turn leads to higher customer satisfaction.