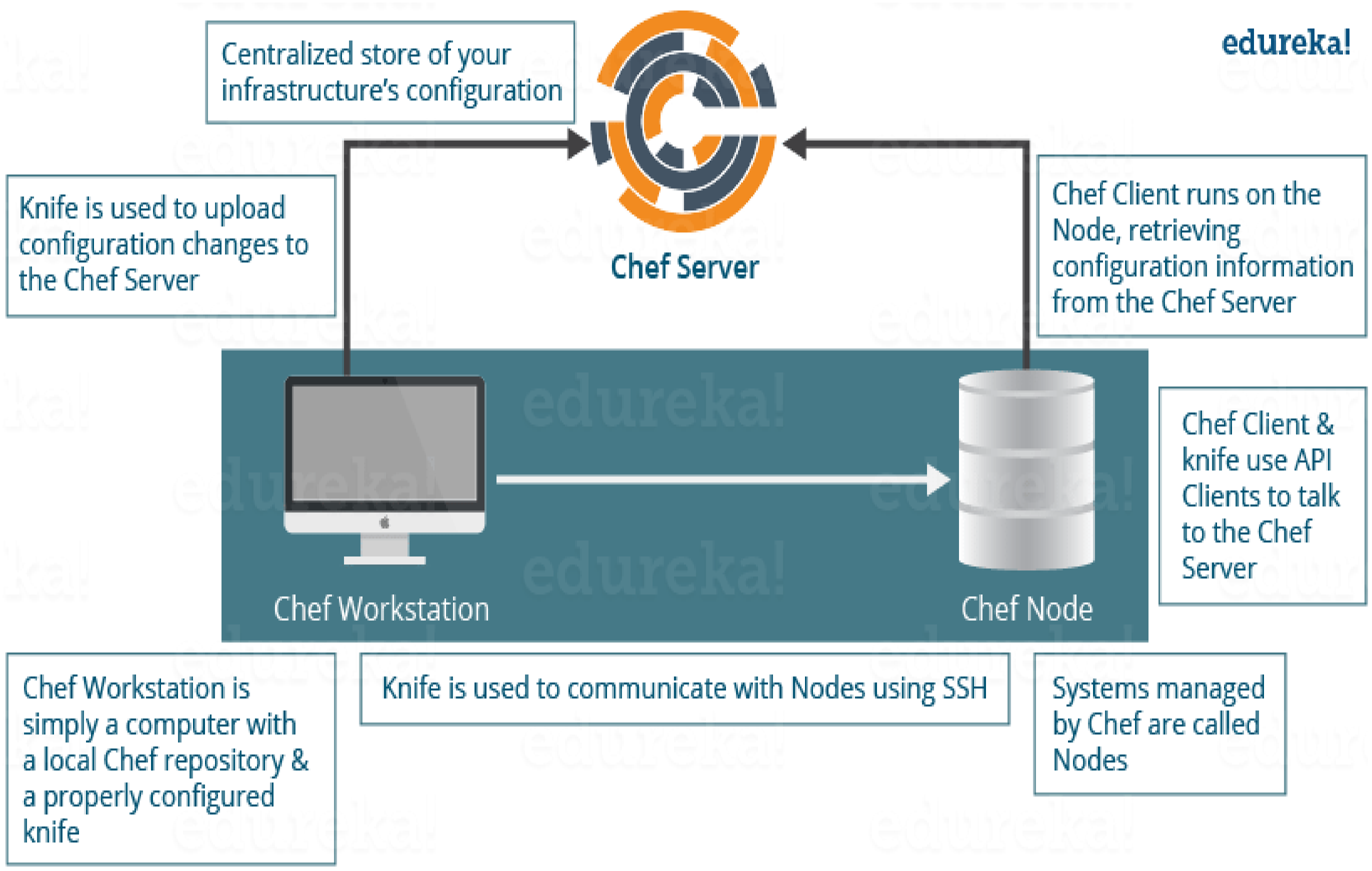
Q1. What is Chef?

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 Architecture Components	
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 & provides search. • It is 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 nodes & contacts the Chef Server for the • information necessary to configure node. • Nodes are sometimes referred as "clients" as they are machines that run the Chef-client software.
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. • Further, it is forwarded to the Chef Server.


Centralized store of your infrastructure's configuration
edureka!
Knife is used to upload configuration changes to the Chef Server
Chef Server
Chef Client runs on the Node, retrieving configuration information from the Chef Server
Chef Client & knife use API Clients to talk to the Chef Server
Chef Workstation
Chef Node
Chef Workstation is simply a computer with a local Chef repository & a properly configured knife
Knife is used to communicate with Nodes using SSH
Systems managed by Chef are called Nodes

Q2. 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.

Q3. What is a Recipe in Chef?

Here also I will suggest you 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.

Q4. 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.

Q5. 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.

Q6. 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:

```
file 'C:\Users\Administrator\chef-repo\settings.ini' do
 content 'greeting=hello world'
```

```
end
```

is same as the below resource:

```
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.

Q7. Are these two Chef recipes the same?

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

&&

```
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.

Q8. 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.

```
service 'httpd' do
action [:stop, :disable]
  end
```

Q9. 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.

Q10. What is run-list in Chef?

My advise is to first explain what is the use of run-list

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.

Q11. 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.

Q12. 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.

Q13. 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.

Q14. 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"**.

Q15. 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.

Q16. 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 instancs.

Q17. 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***.