# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

## SIMULACIÓN DE SERVIDORES DE CORREO, DNS, DHCP MEDIANTE CHEF.

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR EN REDES Y TELECOMUNICACIONES**

**JENNIFER ALEXANDRA ROBALINO ALCIVAR**
jennifer.robalino@epn.edu.ec

**DIRECTOR: ING. FERNANDO VINICION BECERRA CAMACHO, MSc.**
fernando.becerrac@epn.edu.ec

**CODIRECTOR: ING.FABIO MATIAS GONZÁLEZ GONZÁLEZ, MSc.**
fabio.gonzalez@epn.edu.ec

**Quito, enero 2021**

# CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por la Srta Robalino Alcívar Jennifer Alexandra como requerimiento parcial a la obtención del título de TECNÓLOGO SUPERIOR EN REDES Y TELECOMUNICACIONES, bajo nuestra supervisión:

**Fernando Vinicio Becerra Camacho**
DIRECTOR DEL PROYECTO

**Fabio Matías González González**
CODIRECTOR DEL PROYECTO

# DECLARACIÓN

Yo Robalino Alcívar Jennifer Alexandra con CI: 1721866802 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin prejuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

**Jennifer Alexandra Robalino Alcívar**
**Estudiante**

# DEDICATORIA

Dedico este trabajo con todo cariño a mis padres Mireya y Jorge, por el apoyo constante, por llenar mi vida de inspiración, ustedes quienes amo con toda mi existencia y me han permitido trazar mi camino y caminar con mis propios pies.

A mi hermana Erika como siempre ha sido mi cómplice, por ayudarme a crecer, pero sobre todo por nunca cortarme las alas y ser de mis personas favoritas, a mi hermano Terry, por ser tan tú y siempre regalarme risas y las ganas de ser grande, a mi hermano Alex por el apoyo y cariño.

Dedico este trabajo a toda mi familia ¡Que nadie se quede afuera, se los dedico a todos ustedes!

Se lo dedico a Stalin a quien estimo tanto y a quien le debo su apoyo incondicional, por todos esos momentos épicos, por siempre decirme que nunca dude de mi capacidad.

Abigail amiga sincera que siempre me brinda su apoyo, dándome consejos certeros y escuchándome en todo momento.

Se lo dedico a quienes creyeron en mí y para los que no también, porque gracias a todos ustedes estoy logrando culminar un peldaño más de mi vida, para finalizar como no dedicarme a mi este trabajo que me enseño que sin importar cuanto tiempo me tome, todo se puede si de verdad se quiere, un día a la vez.

# AGRADECIMIENTO

Me van a faltar páginas para agradecer a todas las personas que se han involucrado en esta etapa de mi vida profesional, sin embargo, quiero expresar mi gratitud a ese ser supremo que siempre estuvo acompañándome, guiándome, brindándome paciencia y sabiduría hasta estas instancias de mi vida.

Merecen el reconocimiento y gratitud especial, mis padres Mireya y Jorge, quienes son el pilar fundamental en mi hogar, por ese gran esfuerzo y dedicación con la que me ayudan a culminar cada una de mis metas, así como también todos esas enseñanzas y valores que nos inculcan a mis hermanos y a mí.

De igual forma, agradecer a mis compañeros de vida, mis hermanos Alex, Erika y Terry quienes siempre con sus palabras de aliento, ayuda y compañía me hacen sentir orgullosa de lo que soy y de lo que puedo llegar a ser.

Mi profundo agradecimiento a mis abuelitos Jorgito, Gladycita, Briseida y mi angelito en el cielo Luciano, quienes siempre han estado pendiente de la familia y ser partícipes de mi vida, apoyando cada una de mis decisiones y brindándome consejos de vida.

Así mismo, agradezco a toda mi familia que siempre está dispuesta ayudar y a dar una mano en momentos difíciles, a Raúl que se ha convertido en parte de mi familia, por su apoyo incondicional y regalarme conocimiento.

Agradezco a mis amigos Stalin, Guillermo, Anthony y Liliana por las risas y ayuda brindada, al hacer de mi vida universitaria una de las mejores experiencias; como no agradecer Abigail quien siempre ha estado en los momentos más difíciles y por todos los años de amistad brindados.

Agradezco, a mi amigo Adrián por el apoyo y el tiempo invertido, guiándome y brindándome conocimientos, los cuales fueron muy útiles.

De manera especial a mi tutor de tesis Fernando Becerra, por haberme guiado, en la elaboración de este trabajo de titulación, que gracias a sus consejos y correcciones hoy puedo culminar este trabajo.

A la Escuela Politécnica Nacional y a la Escuela de Formación de Tecnólogos, por ser la sede de todo el conocimiento adquirido en estos años.

# ÍNDICE DE CONTENIDOS

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABLAS

# RESUMEN

El presente proyecto de titulación, “SIMULACIÓN DE SERVIDORES DE CORREO, DNS, DHCP MEDIANTE CHEF”, pretende mostrar una introducción a un sistema de automatización nuevo como lo es Chef.

En la sección uno se mostrará una breve introducción sobre detalles generales del proyecto a realizarse como el planteamiento del problema, de la misma forma se detallarán los objetivos y fundamentos que se desarrollarán en el presente proyecto.

En la sección dos se exhibirá la metodología manejada para diseñar el presente proyecto, centrada en el análisis de diferentes aspectos sobre el *software* Chef, que permitirá continuar con el perfeccionamiento de cada objetivo.

En la sección tres se presentarán todos los servicios que proporciona el *software* Chef para una infraestructura; donde Chef hace uso de tres opciones de servicios como lo son Chef *Server*, Chef *Workstation* y Chef *Client*, que se complementan entre sí automatizando la infraestructura dentro de una organización. Posteriormente se realizará la configuración de herramientas necesarias para el arranque de las recetas y el envío de estas hacia Chef *Server*, así como también las configuraciones de una comunicación por medio SSH y el intercambio de claves privadas.

Seguido de la configuración de nodos que ejecutarán sobre sí mismos los servidores de DNS, DHCP y correo, finalizando con los resultados obtenidos en base en las simulaciones de los servidores especificados.

Finalmente, las secciones cuatro y cinco contendrán las conclusiones y recomendaciones obtenidas en el desarrollo de la práctica, continuando con las referencias bibliográficas utilizadas como guías del proyecto de titulación.

**PALABRAS CLAVE:** DevOps, Chef, Servidores, DNS, DHCP, Correo.

# ABSTRACT

*This degree project, "SIMULATION OF MAIL, DNS, DHCP SERVERS THROUGH CHEF", aims to show an introduction to a new automation system such as Chef.*

*In the section one will show a brief on general details of the project, make an introduction such as the statement of the problem, the same way the objectives and foundations that will be developed in this project will be detailed.*

*Section two will show the methodology used to design this project, focused on the analysis of aspects of the Chef software, which will continue with the improvement of each objective.*

*In section three all the services provided by the Chef Software for an infrastructure will be presented; where Chef makes use of three service options such as Chef Server, Chef Workstation and Chef Client, which complement each other by automating the infrastructure within an organization. Subsequently, the configuration of tools necessary to start the recipes and send them to the Chef server will be executed, as well as the configurations of a communication through SSH and the exchange of private keys.*

*This is followed by the configuration of nodes that the DNS, DHCP and mail servers will execute on themselves, ending with the results obtained based on the simulations of the specified servers.*

*Finally, sections four and five contain the conclusions and recommendations obtained in the development of the practice, continuing with the bibliographic references used as guides for the degree project.*

***KEYWORDS:*** *DevOps, Chef, Servers, DNS, DHCP, Mail*

# 1 INTRODUCCIÓN

Con la evolución de los departamentos de TI y el progreso de la gestión en empresas de forma dinámica, con organizaciones que tienen algún tipo de automatización. Se presentan herramientas de DevOps, las cuales permiten la realización de operaciones de forma predeterminada dependiendo del entorno en el que se las aplique, para una mayor flexibilidad y escalabilidad para empresas de diferentes tamaños [1].

Es aquí donde la actual gestión de los servidores de correo, DNS y DHCP presentan un desafío para pequeñas y medianas empresas que no poseen los recursos económicos para adquirir equipos físicos para la gestión de dichos servidores. Debido a la excesiva cantidad de usuarios con tendencia a la saturación de los servicios de telecomunicación. Es en este punto donde se encuentra como una de las soluciones la implementación de dichos servidores en una de las herramientas de DevOps como lo es el *software* Chef, un *software* de gestión de código abierto con una de las arquitecturas más trascendentes. Esta se llevará a cabo como un piloto en un computador, en el que se instalará en entornos virtuales y se descargará los correspondientes paquetes de funcionamiento del *software* Chef para un trabajo multiplataforma de arquitectura Cliente-Servidor, donde los servidores DNS, DHCP y correo podrán desplegarse y llevar a cabo sus funciones [2].

En el **Anexo 1** se adjunta el certificado de funcionamiento el mismo que avala que el presente proyecto ha sido culminado con los parámetros establecidos de simulación.

## 1.1 Objetivo general

Simular servidores de correo, DNS, DHCP mediante Chef.

## 1.2 Objetivos específicos

- Investigar sobre los servicios que ofrece Chef.
- Instalar los servidores mediante Chef.
- Simular los diferentes entornos que se pueden presentar en un ambiente a implementar mediante Chef.
- Verificar los resultados obtenidos con Chef**.**

## 1.3 Fundamentos

### DevOps

La transformación digital en la actualidad ha llegado a los medios de IT, es donde se empiezan a crear entornos de gestión en infraestructuras, es así como llegan las herramientas de DevOps, como sus acrónimos lo dicen *Development Operations* (Desarrollo y Operación) [3].

Mediante estas tecnologías se procura tener una gestión eficaz donde los administradores de los sistemas puedan generar comprobaciones y redundancias para de esta forma mantener activos los sistemas. Ayudando a tener una gestión sin errores de configuraciones manuales de servidor o de usuarios, así es como se crean estas herramientas como Puppet, SaltStack, Chef, entre otras, donde cada una de ellas brindan versiones para los diversos Sistemas Operativos [3].

Todas las herramientas que utiliza DevOps presentan particularidades y similitudes entre ellas; estas se desarrollan bajo licencias de *software* libre, con arquitecturas de modelo Cliente-Servidor y se escriben en lenguajes de línea donde podrán modificarse según las necesidades. Como se puede visualizar en la Tabla 1.1, esto corresponde a un análisis comparativo entre herramientas de DevOps.

Las herramientas DevOps tienen como beneficios la administración combinada, así como la velocidad al desempeñarse en dichas arquitecturas en conjunto con trabajos en la nube, lo que ocasiona que tengan una conexión directa evitando tener un cliente para la ejecución de tareas; de esta manera exista eficiencia en la configuración y administración de los nodos. Además, permite desarrollarse en otras áreas informáticas con el simple hecho de ejecutar un módulo donde este desarrollara el trabajo de administración de archivos, administración y configuración de sistemas, administración de servicios y administración y configuraciones de *backup´s* [3].

**Tabla 1.1** Análisis Comparativo Herramientas DevOps [3]

| Herramientas/ Propiedades | | Chef | Puppet | SaltStack |
|---|---|---|---|---|
| Lenguaje | | Ruby | Ruby | Python |
| Arquitectura | | Cloud y Cliente -Servidor | Cloud y Cliente -Servidor | Cloud |
| Tipo de licencia | | Apache | GNU GLP y Apache | Apache |
| Escalabilidad | | Si | Si | Si |
| Encriptación | | Si | Si | Si |
| Multiplataforma | | Si | Si | Si |
| Sistemas Operativos | | Linux, Windows, OS X, Solaris y IBM AIX | Linux, Windows, Solaris y IBM AIX | Linux, Windows, OS X, Solaris y IBM AIX |
| Plataformas | OpenStack, | Si | Si | Si |
| | VMware | Si | Si | Si |
| | Amazon Web *Service* | Si | Si | Si |
| | Nagios | Si | Si | Si |
| | *Google Computer Engine* | Si | Si | Si |
| | *Team Foundation Server* | Si | Si | Si |

## DevSecOps

La seguridad en el área de TI, representa un papel importante en la configuración, control y el desarrollo de distintas actividades, donde estos deben tener una agilidad, enfoque y posibilidad de respuesta a diferentes entornos.Con la llegada de las herramientas DevOps la gestión de la seguridad es un desarrollo más, con un entorno de trabajo colaborativo se integra DevSecOps, donde el concepto "*Sec*" es la implementación de Seguridad [4].

DevSecOps reintegra la seguridad desde el principio y durante todo el desarrollo de la gestión y control de aplicaciones de determinadas infraestructuras, este permitirá agregar distintas puertas de seguridad impidiendo que se desacelere el flujo de operación, agregando reconocimientos de riesgos, así como también un análisis de este, para posteriormente brindar soluciones y beneficios con relación a lo estudiado.

# 2 METODOLOGÍA

## 2.1 Descripción de la metodología usada

Según el diseño de la investigación se usó la metodología exploratoria, ya que se centró en el análisis de los diferentes aspectos y entornos donde se tuvo las primeras interacciones con *software* Chef [5].Posteriormente, se hizo uso de la metodología descriptiva, la cual maneja una descripción más detallada de los procesos que componen a dichos servicios y finalmente con la metodología cuantitativa se obtuvo respuestas a los procesos realizados de las configuraciones en los servidores DNS, DHCP y Correo electrónico [2].

Con ayuda de bibliografías oficiales se buscó obtener información actualizada sobre el funcionamiento, así como configuración de Chef, adicionalmente se obtuvo las versiones correspondientes en el mercado sobre los distintos servidores de correo electrónico, DNS y DHCP.

Para obtener la información necesaria en cuanto a instalación se realizó una investigación previa de las versiones que sean compatibles al ordenador en el que se trabajó, con la finalidad de identificar cómo se adapta en los entornos planteados por los servidores, la cual se verá enlazada a un *software*.
A parte de utilizar versiones compatibles para el ordenador, se realizó un análisis de los requerimientos necesarios en cuanto al montaje de los servidores en un entorno de virtualización [6].

Para estas instalaciones es posible considerar una virtualización total, o a nivel de sistema operativo. Cada una de las características permitió la automatización de la infraestructura del sistema para implementarse, esto admitió el despliegue de los diferentes servidores [7].

Una vez instalados todos los requerimientos necesarios, se procedió a la implementación de cada uno de los servidores dentro de Chef, para el servidor DNS se simuló repositorios centrales de información para las traducciones de los nombres de dominio, de la misma forma se generó el entorno para DHCP con el fin de utilizar un rango de direcciones IP para una asignación dinámica. Por consiguiente, con los

servidores anteriormente descritos se implementó un servidor de correo electrónico para la transmisión y recepción de mensajes.

Una vez implementada la simulación mediante Chef de los diferentes servidores se realizaron las pruebas necesarias para la verificación del correcto funcionamiento del servidor DNS, DHCP y correo electrónico, de esta manera descubrir si hay un posible error y corregirlo si fuera el caso.

# 3 RESULTADOS Y DISCUSIÓN

El entorno de trabajo desarrollado permitirá la simulación de servidores como DNS, DHCP y correo mediante Chef, el cual ayudará a entender de forma más clara como es el trabajo de automatización por medio de una herramienta DevOps. Esto se va a lograr mediante el conocimiento de cada una de las características que ofrece el *software* Chef. Una vez identificados los distintos servicios que proporciona Chef, se realizará la implementación de los servidores en una infraestructura determinada; además, se procederá a realizar las distintas simulaciones de cada uno de los entornos. Todo esto se verificará a través de los resultados obtenidos con Chef.

## 3.1 Investigación de los servicios que ofrece Chef

Con ayuda de bibliografías oficiales se buscó información actualizada sobre el funcionamiento, así como la configuración de Chef, adicionalmente se investigó como obtener las versiones y configuraciones sobre los distintos servidores de correo, DNS y DHCP.

Por otra parte, el sistema operativo utilizado en el presente proyecto de investigación Ubuntu18.04 opera con las versiones más estables de los servidores DNS, DHCP y correo con el objetivo de garantizar su correcto funcionamiento, estas versiones se seleccionan dentro de la misma plataforma virtual del software chef denominada "*chef supermarket"* la cual es una base datos en la que se almacenan todas las versiones compatibles con este sistema.

### Chef

En enero del año 2009 fue creado el *software* Chef por Adam Jacob, con el objetivo gestionar configuraciones en el campo empresarial de los centros de datos de Amazon y Microsoft, suministrando así una herramienta de *consulting* que esté abierta a los intereses y exigencias del momento. Tomando el nombre Chef a razón de que se pueden crear módulos en forma de recetas; destacando la atención de forma sustanciosa a las empresas [1].

Chef ha ido actualizando y desplegando distintas plataformas y actualizaciones para un mejor rendimiento de *software* [8]. Es imprescindible saber que este s*oftware* dirigido a configuraciones de *management* fue y es escrito en los lenguajes de programación Erlang y Ruby, donde estos lenguajes ayudan a la rapidez del mantenimiento y configuración de los servidores de una organización [8]. Es uno de los *software* más

flexibles y cuenta con un soporte óptimo, convirtiéndose en una de las opciones más destacada para grandes empresas.

Este *software*, al ser multiplataforma, posee como característica principal el funcionamiento con herramientas de entornos virtuales, por la misma razón que permite que este sea escalable en cualquier ambiente.
La Arquitectura Chef trabaja con un sistema de gestión principal que es un servidor "Chef *Server*" y estaciones de trabajo llamadas "*Workstation*", donde la estación de trabajo entabla una comunicación directa con el servidor y este con los nodos, de esta forma se ejecuta de manera jerárquica [9]. Con la creación de libros de cocina (*cookbook*) en las unidades de trabajo. Posteriormente ser evaluadas por comandos como cuchillos que se agregaran en el servidor Chef he interactuado con el mismo mediante este se podrá administrar nodos, recetas, atributos roles y diversos ambientes, como se puede ver en la Figura 3.1

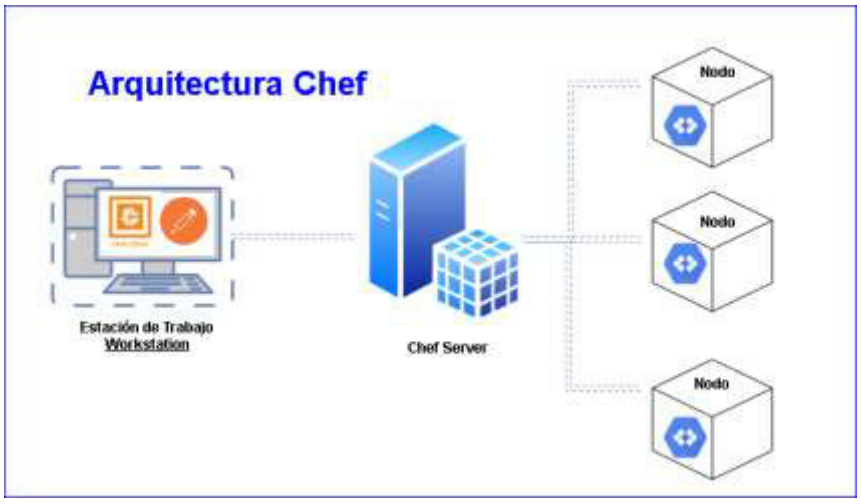


**Figura 3.1** Arquitectura Chef

Chef presenta diferentes tipos de *software* de gestión dependiendo del escenario que el usuario quiera proporcionar a su entorno, al ser un *software* de código abierto que brindará algunos entornos como Chef Infra *Server*, Chef *Workstation*, Chef InSpec, Chef Habitat y Chef Automate [9].

### Chef Infra *Server*

Este equipo hace las funciones de centro de datos para toda la configuración, es decir, almacena todos los libros de recetas, así como las diferentes políticas que se aplican a los clientes. De igual manera proveen a los nodos de todas las distribuciones de

archivos, hay que recalcar que el servidor va muy unido a las operaciones que realiza Chef infra *client*, ya que este último configurará todas las especificaciones provistas por el servidor en los nodos [10].

**Tipos de Chef *Server*:**

- ***Open Source Chef Server*:** Esta es una de las versiones que está dirigida para pequeñas y medianas empresas, trabajando con código abierto, se podrá gestionar diversos módulos, al ser gratuita proporcionará algunas limitaciones.
- **Servidor de Chef alojado:** Este servidor es usado más para la parte comercial como principal característica, donde se encontrará en repositorios efectivos que ayudarán a resolver problemas y brindará soluciones predeterminadas.
- **Chef Privado:** Este tipo de servidor presenta herramientas más completas y se adaptará a los requerimientos de los clientes en específico.

### Chef Infra *Client*

Chef-*Client* opera como un agente que se ejecuta de manera autónoma en todos los nodos registrados en Chef Infra *Server*, su principal función es realizar el procedimiento adecuado para llevar a cada nodo a un estado en concreto como los que se presentan a continuación [11]:

- Registro adecuado como la autenticación de los nodos en el servidor.
- Compilar todos los recursos como recetas y el resto de las dependencias.
- Tomar las acciones para configurar las recetas en el nodo.
- Comunicación continua del estado hacia Chef Automate.
- Realizar la sincronización de los libros de cocina desde el servidor hacia los nodos.

Como clave en la infraestructura de Chef los nodos también poseen una clasificación:

- **Nodos Virtuales:** Estos se encontrarán en dispositivos virtuales.
- **Nodos instancias de Cloud:** Se localizan en entornos de nube de acuerdo con sus componentes.

- **Nodos Físicos:** En esta clasificación se tendrán el nodo en cualquier dispositivo físico como enrutadores, tablets, puntos de trabajo y entre otros dispositivos

### Chef *Workstation*

Esta versión es donde se realizarán las configuraciones principales de los libros de cocina, esta estará ubicada en la máquina local, además de brindar los servicios de ejecución remota por medio de redes ad hoc, escaneo remoto de los clientes lo que lo convierte un *software* sólido para pruebas [12].

### Chef InSpec

Esta herramienta es usada para realizar auditorías a las aplicaciones e infraestructura por medio de comparaciones del estado actual del equipo con el estado esperado haciendo que los códigos utilizados sean fáciles tanto de leer como de escribir. Adicionalmente muestra informes sobre el estado y detecta las posibles infracciones que se puedan suscitar [1].

### Chef Habitat

Tiene como principal característica centrar la configuración entorno a las propias aplicaciones, es decir, automatizará la aplicación para no depender de la infraestructura; esto es principalmente útil al utilizar Chef en máquinas virtuales.

La capacidad de esta herramienta de construir e implementar en cualquier lugar permite que la gestión de las aplicaciones sea de una forma mucho más sencilla, la manera en la que se lograría la automatización de estas aplicaciones es empaquetar todos los recursos de estas, para poder ejecutarse de manera eficiente garantizando que todas las dependencias estén disponibles [13].

### Chef Automate

Es un conjunto de herramientas que brindan una automatización del trabajo para con esto lograr una implementación continua en el sistema, adicionalmente provee de una interfaz gráfica para el usuario la cual presentará todos los nodos conectados al servidor y sus respectivos estados.

Adicionalmente recoge los datos de las configuraciones en tiempo real para que los mismos se puedan filtrar, el funcionamiento es paralelo al trabajo realizado por Chef

Management, ya que ambos permiten una colaboración de los equipos y validan los entornos de desarrollo [14].

### Bloques de construcción clave de Chef

Chef trabaja bajo una estructura ideal que permite usar diferentes bloques de construcción para la administración y automatización, estos son los ejes principales para que toda infraestructura se desarrolle [15], estos bloques se dividen en tres ejes principales que son:

- **Libros de Cocina.**
  El *Cookbook* es una recopilación de recetas, este es el bloque de contrición principal llevando toda la información trabajada desde la estación de trabajo hacia Chef *Server*. Esta presentará recetas en su interior que obtendrán información relevante para una infraestructura llevando datos para su posterior ejecución [16].

- **Receta.**
  Es aquella que lleva en su interior un conjunto de datos llamados atributos que se utilizan para la administración de una infraestructura. Estos atributos son aquellos que cambian el estado actual o existente de un nodo en particular. Las recetas se cargan a Chef Client cuando se ejecutan en el nodo, compartiendo la información de los atributos para llegar a un estado definido de cada recurso implementado.

- **Recurso.**
  Es la unidad básica de una receta con diferentes tipos de estados. En una receta puede existir varios recursos lo que ayudará a configurar y administrar la infraestructura, existen recursos que tienen ficheros creados por defecto llamados default.rb. En la Tabla 3.1 se presenta los recursos principales de la estructura de un libro de cocina [15].

**Tabla 3.1** Estructura predeterminada del libro de recetas.

| Recurso | Función | Extensión |
|---|---|---|
| Attributes | Contiene valores clave con distintos parámetros y niveles. | .rb |
| Template | Sustituye un valor de atributo, para crear la versión final del archivo en el nodo. | .erb |
| Metadata | Gestiona la ejecución y garantiza que todas las piezas se transfieran al nodo, incluye información detalles del paquete. | .rb |
| README | Archivo que contiene información compactada he importante sobre el sistema. | .md |
| Recipes | Gestiona el contenido de un archivo en el nodo | .rb |

## Conceptos Básicos en Chef

Algunos de los nombres en Chef están relacionados a implementos de cocina, los mismos que poseerán una determinada caracteriza y ejecución, de esta manera es de importancia conocer algunos de los conceptos claves como los que se presenta a continuación:

- **Nodo:** Es todo servidor virtual o físico que mantiene una configuración en relación con Chef-*Client*.
- ***Workstation*:** Este es el equipo que está específicamente configurado para mantener una sincronización con Chef-*Server*, manejando distintos repositorios para la asignación de roles.

- ***Cookbooks*:** Es la unidad que poseerá recetas, es decir, configuraciones y las políticas que precisarán el entorno de trabajo y distribución de los sistemas a implementarse, con valores de atribución, extensiones. También conocidos como bibliotecas de recursos y plantillas de recetas [16].
- **Recetas de cocina:** son aquellas que poseen especificaciones de los recursos en líneas de un lenguaje de programación (Ruby), estas presentarán de forma ordenada dependiendo de su especificación y aplicación.
- ***Knife*:** El llamado cuchillo es la herramienta esencial de línea de comando, que facilitará la realización de labores como la implementación de una interfaz que ayudará a la administración de nodos, recetas, libros de cocina, roles y entre otras funciones de gestión [17].
- **Rol:** Es la función que cumple una receta para desempeñar o ejecutarse en el nodo.
- **Run-List:** Es la ejecución de una lista que posee un rol en específico (recetas) dependiendo del orden en que se disponga la ejecución.
- **Ruby:** Es un lenguaje de programación de código abierto, que permitirá la realización de la sintaxis de las recetas de cocina de manera que proporcione la facilidad de interpretar y escribir [18].

### Descripción de la herramienta de virtualización

La virtualización es una de las herramientas principales para el despliegue de distintos entornos de trabajo de prueba en base a programas de computadoras que permiten realizar pruebas de fenómenos reales en un mundo virtual, en las que se puede cambiar variables y parámetros para verificar una hipótesis. Para así, según los resultados obtenidos, tomarlos como aciertos o realizar corrección de errores; de esta manera es como las simulaciones virtuales se llevan un papel protagónico.

En este proceso como principal actor es una máquina virtual creada en un servidor que se encuentra alojado en la Escuela Politécnica Nacional que lleva el sistema operativo Windows 10 debido a que cuenta con una interfaz más sencilla y conocida, en esta se realizaron las correspondientes configuraciones de puertos de red, memoria y disco. Sin olvidar que se active la función de virtualización asistida por *hardware* donde esta opción permitirá realizar máquinas virtuales dentro del entorno ya creado, el entorno en de esta máquina virtual debe cumplir con requerimientos, como se indica en la Figura 3.2 y la Figura 3.3.

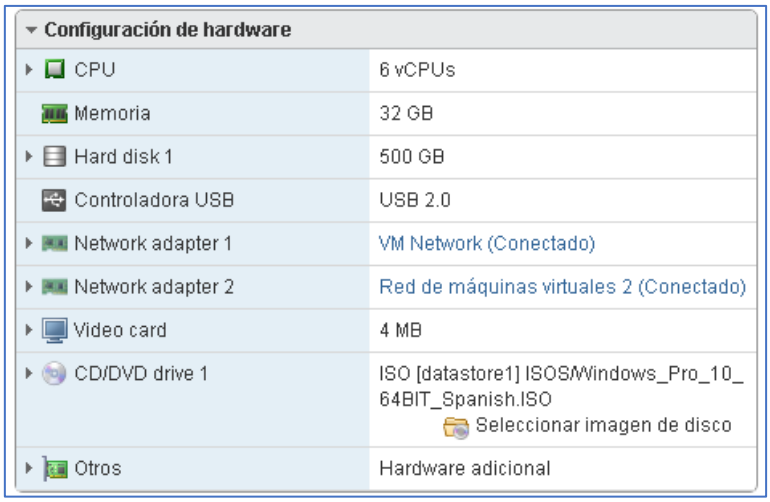


**Figura 3.2** Configuraciones de *hardware*

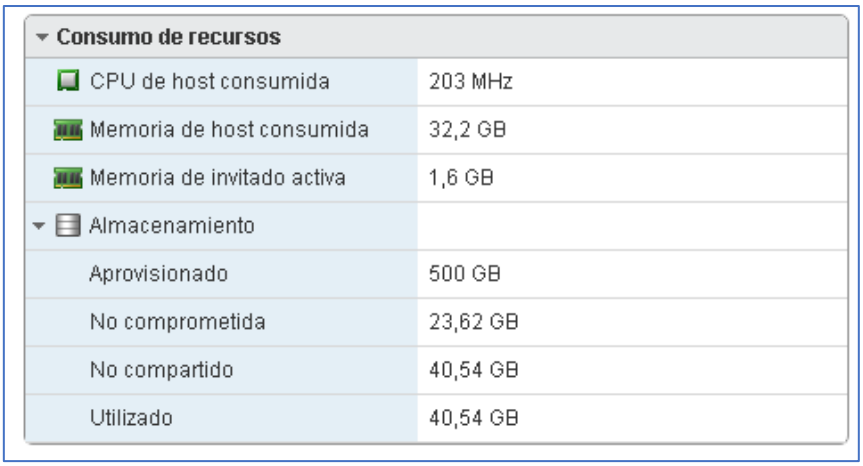


**Figura 3.3** Consumo de recursos

### VMware

VMware es un *software* de virtualización, el cual admite que se ejecuten diferentes sistemas operativos en una única computadora, es decir, permite establecer diferentes entornos en un escritorio sin la necesidad de reiniciar el equipo principal. Este *software* trabaja mediante interoperabilidad entre sistemas operativos, de forma que cada sistema operativo se encuentre aislado para proteger su respectivo entorno [19].

El aislamiento de los sistemas operativos puede ser rentable para brindar soluciones a los departamentos de TI, los cuales pueden realizar pruebas de aplicaciones [19].

El *software* VMware se instala en la máquina virtual de Windows 10, donde se montarán las máquinas virtuales del servidor, estación de trabajo y nodo cliente con el

sistema operativo Ubuntu 18.04; como primer punto se requiere la descarga de las imágenes ISO estás se las puede encontrar en los sitios oficiales de descarga.

Como se aprecia en la Figura 3.4 se lleva a cabo la instalación del *software* de virtualización VMware *Workstation* Pro 16.

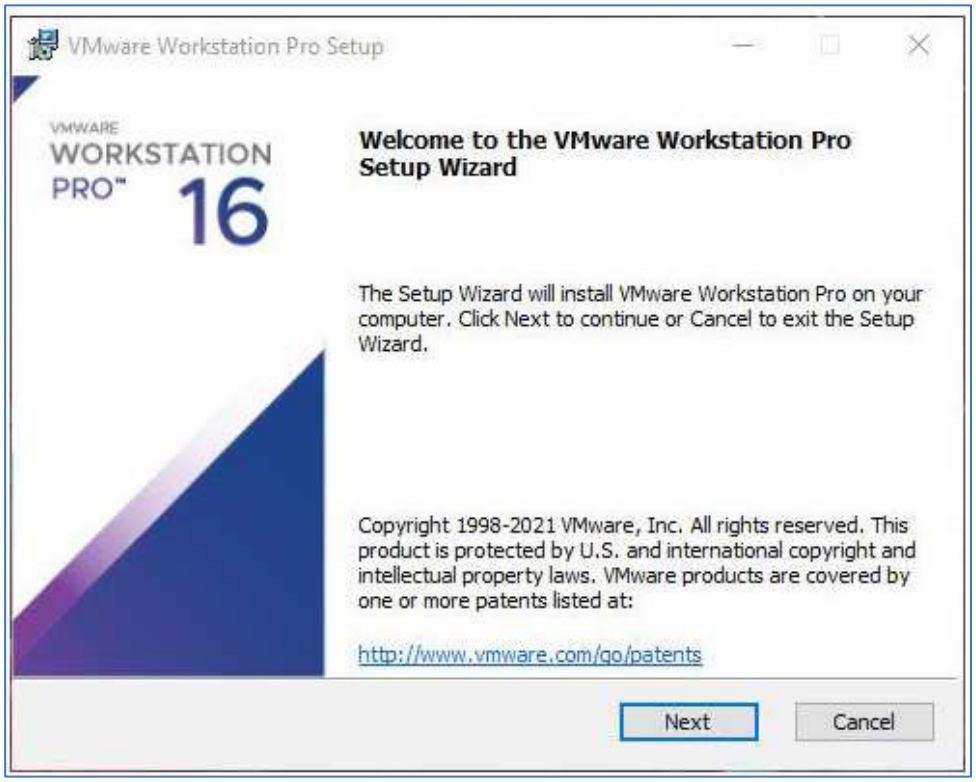


**Figura 3.4** Instalación *software* VMware

## 3.2 Instalar los servidores Chef

Para poder realizar una simulación efectiva del sistema Chef haciendo uso de los sistemas operativos de *software* libre Linux se debe contar con 3 componentes primordiales los cuales son:

- Chef *server*
- *Workstation*
- *Client node*

Se prosigue al montaje de la primera imagen ISO correspondiente es el servidor de Ubuntu 18.04 desde el cual correrá el servidor Chef, este montaje tendrá las siguientes especificaciones:

- Memoria RAM de 4 Gb
- Capacidad de procesamiento doble

- Capacidad de almacenamiento de 20 Gb
- Adaptador de red tipo NAT para lograr conectividad dentro de la red interna del servidor.

Del mismo modo la estación de trabajo, así como el nodo cliente contará con las mismas especificaciones para la correcta ejecución del sistema operativo que en el caso de la *Workstation* y el "*client node*" será el Ubuntu 18.04 LTS, las máquinas creadas se muestran en la Figura 3.5.

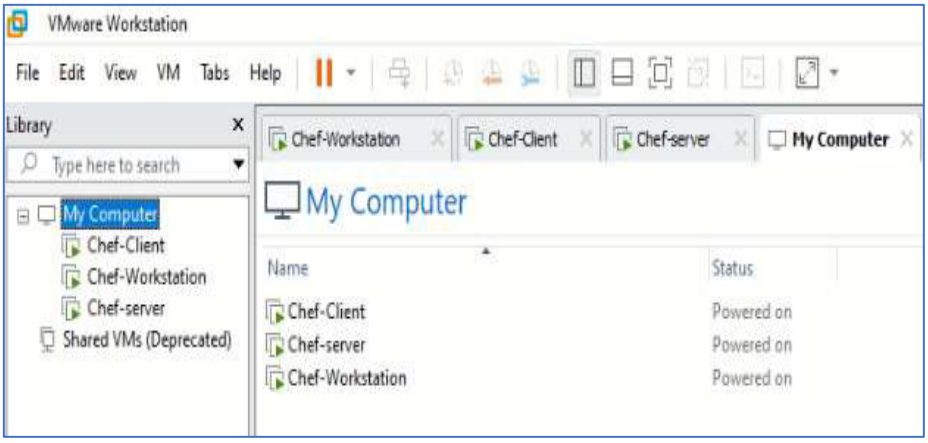


**Figura 3.5** Máquinas virtuales creadas

Durante la instalación de cada uno de los sistemas operativos se debe configurar correctamente el nombre de usuario y el nombre del equipo correspondiente, cada una de las máquinas poseerá sus propias credenciales. A continuación, se procede a verificar cada una de las direcciones con las que cuentan las máquinas, esto se realiza con las siguientes líneas de comandos como se puede apreciar.

```
sudo apt install net-tools
ifconfig
```

El primer comando presentado instalará el paquete que contiene todas las herramientas para la ejecución del siguiente comando, el cual mostrará en pantalla los puertos de red donde se puede observar la dirección que fue asignada al equipo, esto se visualiza en la Figura 3.6.

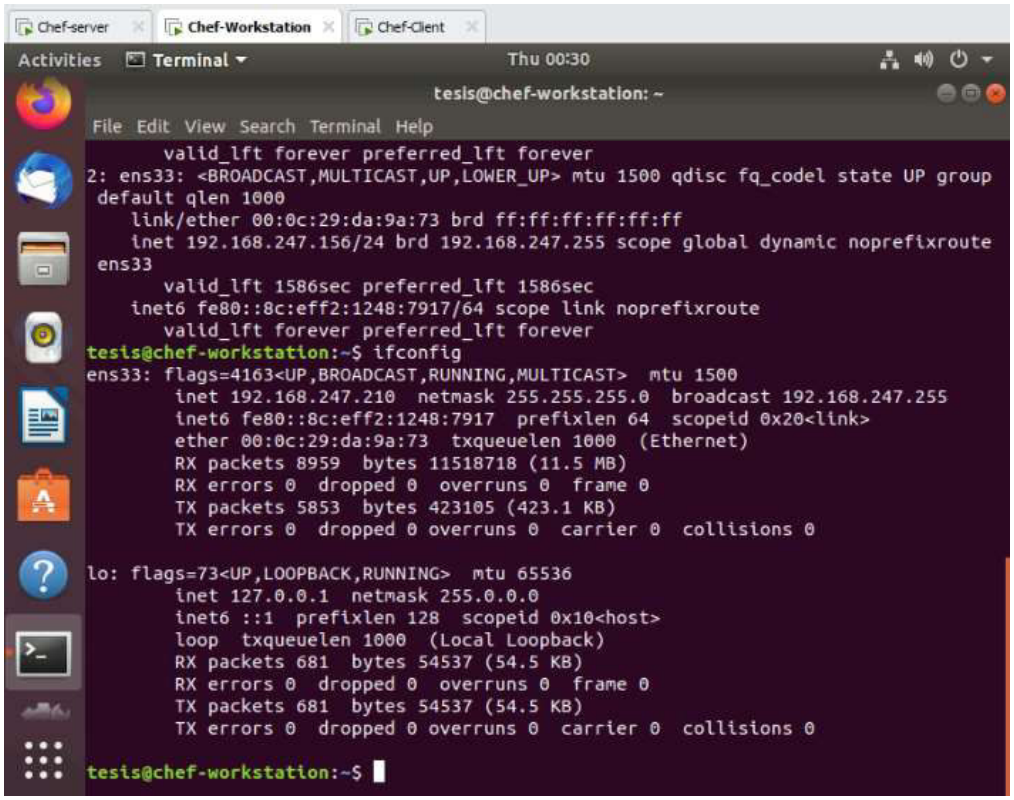


**Figura 3.6** Comandos de verificación de puertos de red

A continuación, una vez conocidas las direcciones y establecido los nombres de las máquinas, se editará el archivo hosts en el directorio /etc/hosts, para realizar esta acción se utiliza el siguiente comando:

```
sudo nano /etc/hosts
```

Los parámetros por ingresar dentro de este archivo son los siguientes:

- **Chef *Server***
  - IP: 192.168.247.209
  - Nombre máquina: chef-*server*
- ***Workstation***
  - IP: 192.168.247.210
  - Nombre máquina: chef-*workstation*
- **Nodo Cliente**
  - IP: 192.168.247.211
  - Nombre máquina: chef-*client*

Una vez modificado el archivo se procede a realizar los cambios como se muestra en la Figura 3.7, tanto para la estación de trabajo como para el nodo cliente, esto se realiza para que exista comunicación entre los tres equipos.

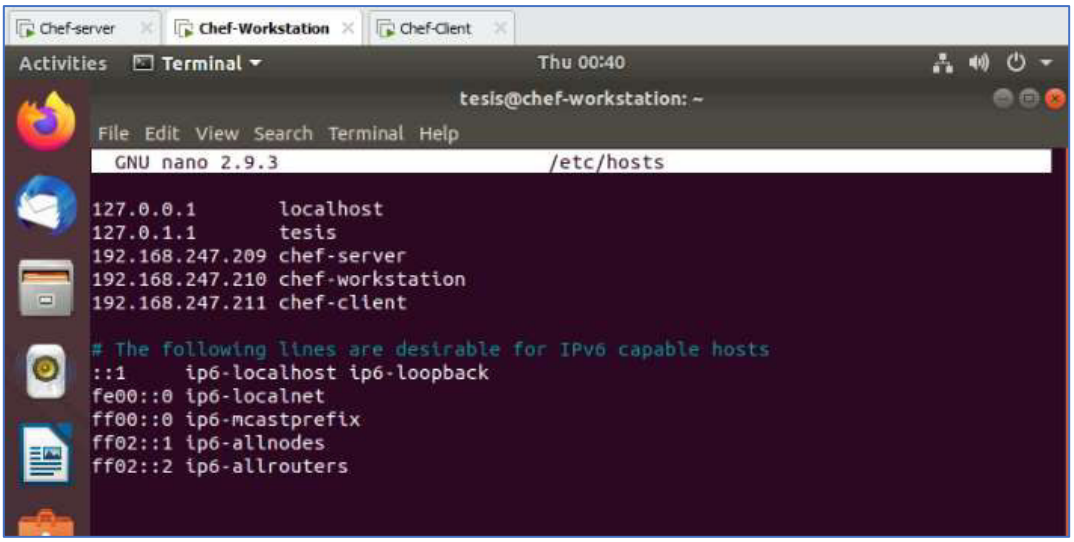


**Figura 3.7** Configuración archivo hosts

Se realiza una verificación, donde se observará que el usuario está correctamente creado.

```
hostname -f
```

Se prosigue a ejecutar el comando de actualización en las tres máquinas virtuales, este se actualizará presentando el siguiente mensaje, esto se puede visualizar en la Figura 3.8.

```
sudo apt update
```

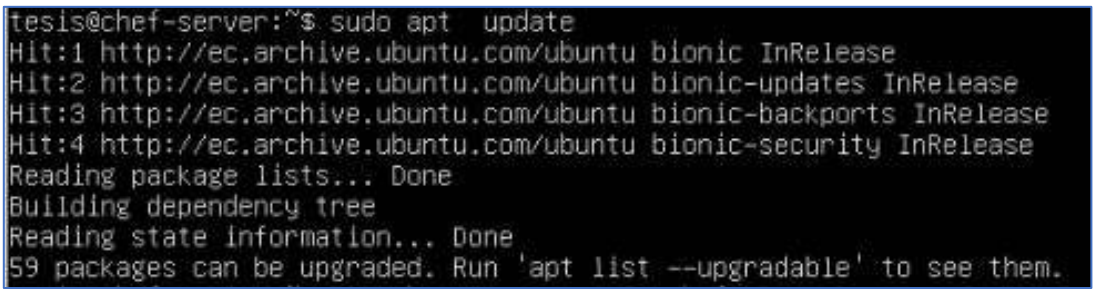


**Figura 3.8** Actualización de sistemas

En el **Anexo 2** se mostrará el listado de todos los comandos ingresados en el servidor Chef, que fueron previamente explicados.

### Instalación y configuración de Servidor Chef

El Servidor Chef se instalará en Ubuntu *Server* 18.04, donde este componente de arquitectura entablará comunicación con la *Workstation* y los nodos clientes.

Luego se procede a ejecutar el comando que descargará el paquete Chef-*Server*, para esto se debe conocer la versión de Chef- *Server* que sea compatible con la versión de Ubuntu 18.04, se presentará la descarga como se observa en la Figura 3.9.

```
wget https://packages.chef.io/files/stable/chef-server/13.0.17/ubuntu/18.04/chef-server-core_13.0.17-1_amd64.deb
```

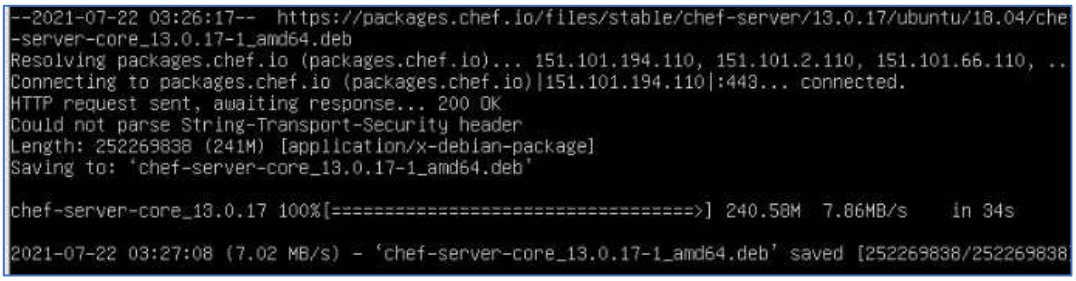


**Figura 3.9** Descarga del paquete chef-*server*

Posteriormente se requiere ejecutar el comando que permitirá instalar el paquete del servidor.

```
sudo dpkg -i chef-server-core_*.deb
```

Iniciar los servicios de Chef-*Server* mediante la siguiente línea de comando que pertenece a la utilidad que proporciona Chef.

```
sudo chef-server-ctl reconfigure
```

Se verifica el estado de los servicios instalados utilizando el siguiente comando, el cual se indica en la siguiente Figura 3.10.

```
sudo chef-server-ctl status
```


**Figura 3.10** Verificación de estados de servicios

En el servidor se precisa la instalación de NTP o *Network Time Protocol* para la sincronización del reloj de servidor, esta instalación es de suma importancia, ya que el sistema Chef es perceptivo a las alteraciones de reloj, de esta forma es como se requiere la ejecución de las siguientes líneas de código.

```
sudo apt install ntp
```

Se prosigue a correr la línea de código, seguidamente se ingresa el comando la interfaz pedirá la clave del servidor para proseguir con la ejecución.

```
systemctl start ntp
```

Se continúa verificando que la sincronización esté ejecutada, de la misma forma para que se inicie automáticamente NTP en el instante del arranque se ejecuta las siguientes líneas de código, donde se mostrará que la autenticación fue completada, tal como se indica la Figura 3.11.

```
timedatectl
systemctl enable ntp
```

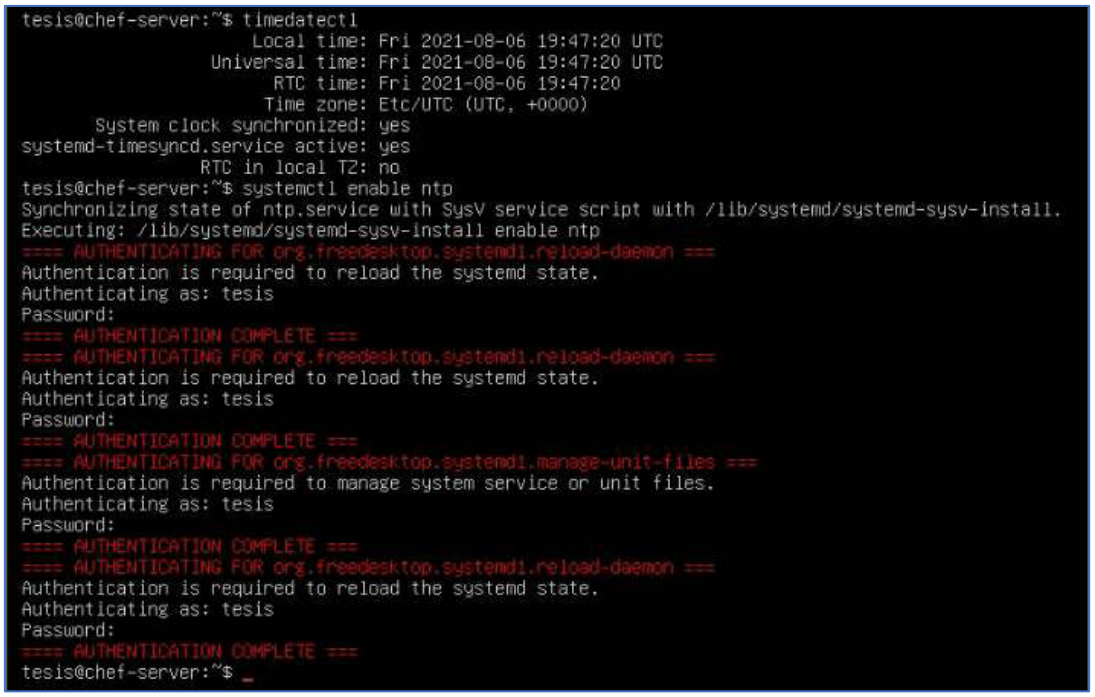


**Figura 3.11** Autenticación NTP

**Instalación Chef Manage.**

Chef Manage o también conocido como la consola de administración Web del servidor, es donde el usuario administrador podrá visualizar en una interfaz gráfica, la gestión de atributos, parámetros de las listas de ejecución, entornos, *cookbook* y roles, entre otras opciones que ayudarán a la gestión del servidor de manera más sencilla.

Este se inicia con la instalación o ejecución de la siguiente línea de código.

```
sudo chef-server-ctl install chef-manage
```

Seguido del comando de instalación se requiere reconfigurar el servidor Chef con la ejecución del siguiente código.

```
sudo chef-server-ctl reconfigure
```

Por consiguiente, configurar el Chef Manage con la siguiente línea de comando, este permitirá la aceptación de las licencias durante la respectiva ejecución.

```
sudo chef-manage-ctl reconfigure
```

Sucesivamente se efectúa con la verificación del Chef Manage ingresando a un navegador, ingresar la dirección IP del servidor, se presentará la interfaz Web como se

indica en la Figura 3.12, para posteriormente cuando se cree el usuario y organización se presente en la misma.

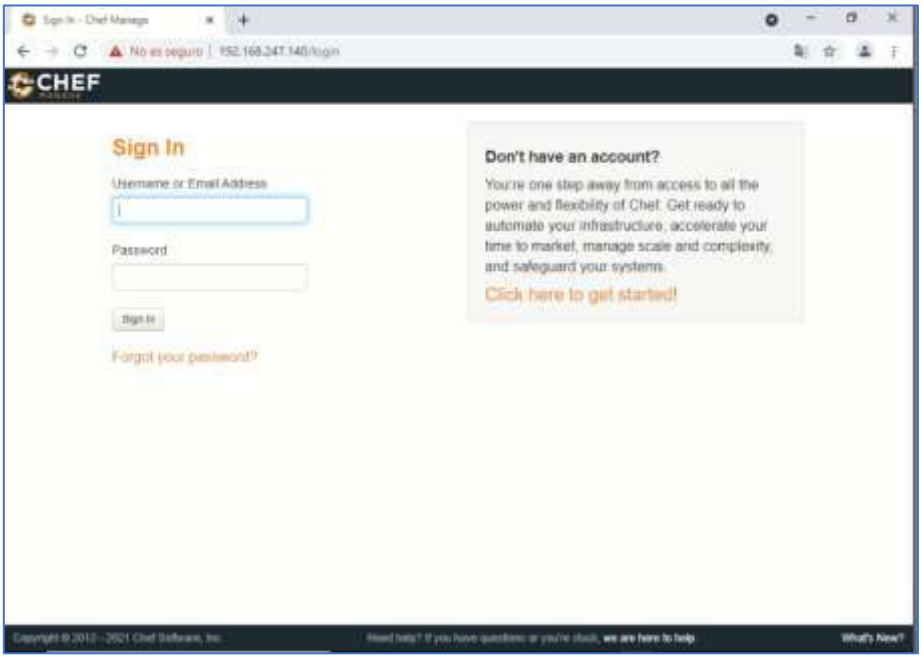


**Figura 3.12** Chef *Manage*

**Creación de Usuarios y Organización**

Para poder entablar una comunicación entre las distintas máquinas de la arquitectura de Chef presentada, se requiere la creación de un usuario y una organización que contará con sus respectivas claves.

Se prosigue a la creación de un directorio el cual tendrá el nombre **.chef** este almacenará las claves, este se creará en el directorio de inicio.

```
tesis@chef-server:~$ mkdir .chef
```

Crear el usuario con el siguiente comando, en el mismo se especificará el nombre de la extensión **.pem** que es la clave RSA, para las especificaciones del usuario se usa los siguientes ajustes:

- **Usuario:** chefadmin
- **Nombre:** Chef
- **Apellido:** Tesis
- **Correo:** chefadmin@tesis.com
- **Contraseña:** Se ingresará la contraseña que el usuario desee.
- **Clave RSA:** chefadmin.pem

```
sudo chef-server-ctl user-create chefadmin chef tesis
chefadmin@tesis.com '*******' --filename ~/.chef/chefadmin.pem
```

A continuación, en la Figura 3.13 se puede ver como se ingresa la siguiente línea de código para la verificación de la creación del usuario en el Chef *Server*.

```
sudo chef-server-ctl user-list
```


**Figura 3.13** Comprobación de creación de usuario

Para la creación de la organización ingresará el siguiente comando que contendrá la siguiente información:

- **Organización:** chef-org
- **Nombre Organización**: Tesis Chef *Infrastructure*
- **Clave RSA:** chef-org.pem

```
sudo chef-server-ctl org-create chef-org "tesis chef infrastructure"
--association_user chefadmin --filename ~/.chef/chef-org.pem
```

Se realiza la ejecución del comando de comprobación para ver la lista de organizaciones creadas en el servidor Chef, como se indica en la Figura 3.14.

```
sudo chef-server-ctl org-list
```

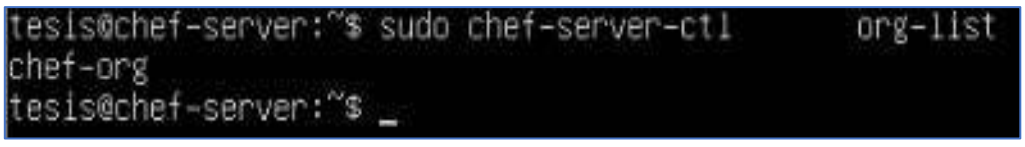


**Figura 3.14** Verificación de creación de organización

Se prosigue a verificar que el usuario creado es correcto, dirigirse al Chef *Manage* o interfaz Web, donde se debe ingresar el usuario y contraseña como se observa en la Figura 3.15, cuando se logre ingresar e iniciar sesión en la misma se podrá verificar que dentro de esta también está la organización creada anteriormente como se muestra en la Figura 3.16.

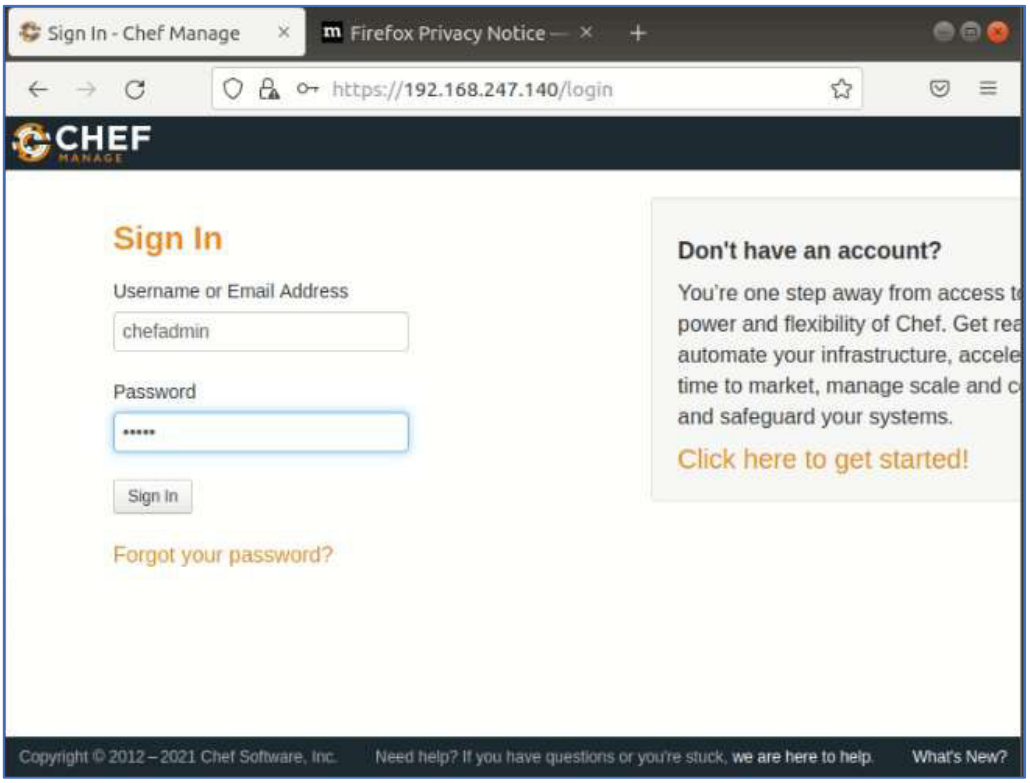


**Figura 3.15** Verificación de usuario en Chef Manage

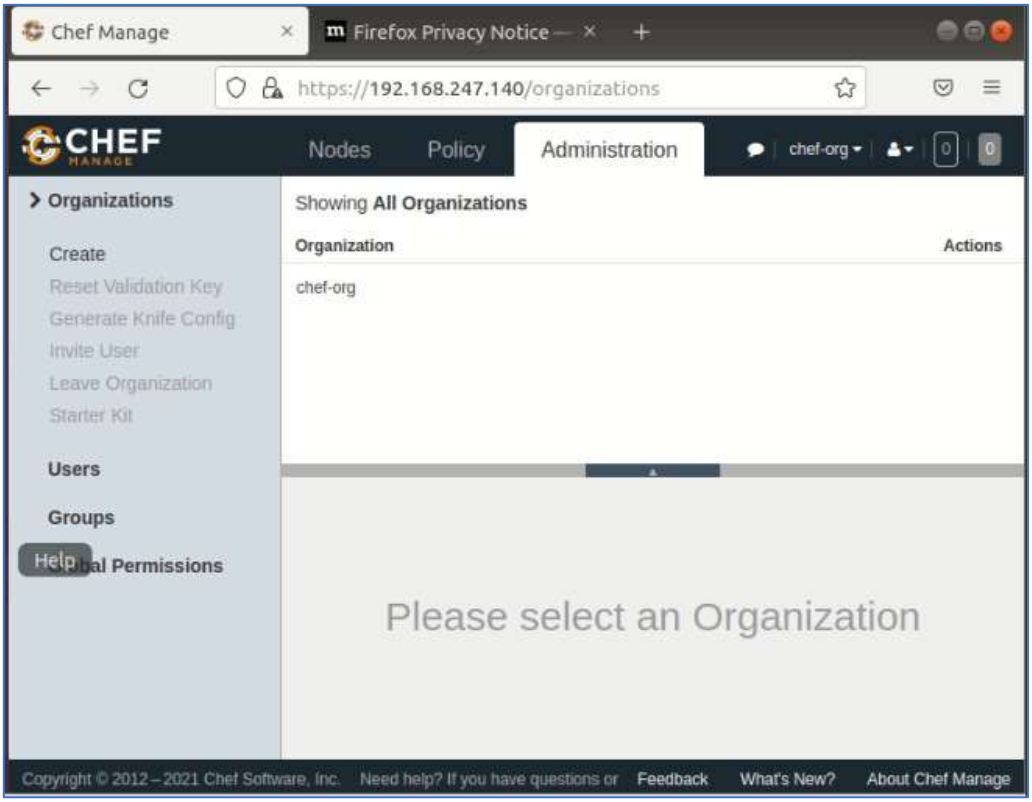


**Figura 3.16** Verificación de organización en Chef Manage

Subsiguiente a la creación de usuarios y organización se instalará un administrador de Chef en el cual se podrá visualizar informes de los procesos que se van creando o generando en la organización, así como también en los nodos, estos informes se ejecutan con la siguiente línea de código.

```
sudo chef-server-ctl install opscode-reporting
```

Se continúa con la reconfiguración del servidor ejecutando el siguiente código.

```
sudo chef-server-ctl reconfigure
```

Se realiza la configuración de los entornos de informes ejecutando el siguiente parámetro.

```
sudo opscode-reporting-ctl reconfigure
```

Realizada la instalación de opscode se ingresa la siguiente línea de código para la verificación de funcionamiento y de los complementos instalados, si ingresado este comando no se ejecuta ningún problema se podrá continuar con las configuraciones.

```
test opscode-reporting-ctl
```

De manera similar, se instalará trabajos Chef *Push* perteneciente a opscode que permitirá efectuar trabajos del servidor contra nodos independientes de la ejecución de **chef-*client*.**

```
sudo chef-server-ctl install opscode-push-jobs-server
```

Se volverá a reconfigurar el servidor con la siguiente línea de código.

```
sudo chef-server-ctl reconfigure
```

Ejecutando el siguiente código se configurará el módulo *Push* Jobs.

```
sudo opscode-push-jobs-server-ctl reconfigure
```

### Instalación y configuración puesto de trabajo (*Workstation*)

Esta será la máquina donde se realzará las configuraciones de los distintos atributos para la administración de los diferentes nodos, para esta máquina se usará Ubuntu 18.04 con interfaz gráfica donde se ingresará comandos mediante la consola.

Se descargará la estación de trabajo con la versión compatible a Ubuntu 18.04 con la siguiente línea de código, como se observa en la Figura 3.17.

```
wget https://packages.chef.io/files/stable/chef-workstation/
0.2.43/ubuntu/18.04/chef-workstation_0.2.43-1_amd64.deb
```

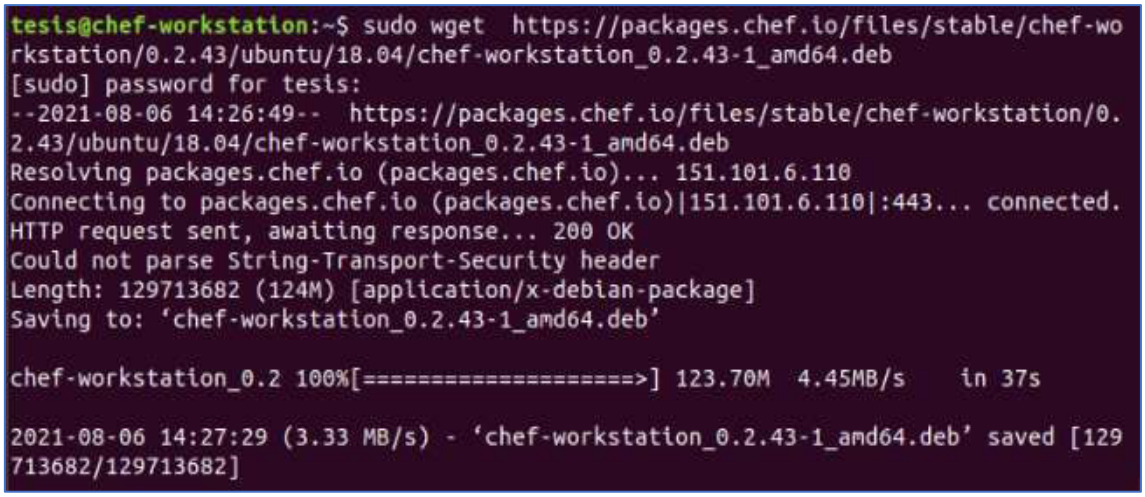


**Figura 3.17** Instalación *Workstation*

Se instala una estación de trabajo con la siguiente línea de código.

```
sudo dpkg -i chef-workstation_*.deb
```

Por consiguiente, se provee la creación del repositorio de Chef. Este tendrá toda la información relacionada con los *cookbooks*, recetas y entre otros archivos. También se prosigue a crear el directorio donde se guardarán las claves RSA.

```
tesis@Chef-worstation:~$ chef generate repo chef-repo
tesis@Chef-worstation:~$ mkdir ~/chef-repo/.chef
tesis@Chef-worstation:~$ cd chef-repo
```

Se continúa con la generación de claves RSA, las mismas que se deben realizar a dirigiéndose al directorio **chef-repo**.

**Creaciones claves RSA**

Se ejecutarán las claves RSA con el siguiente comando para posteriormente entablar un acceso con el servidor, esta creación de claves va a autenticar la estación de trabajo y se presentará como en la siguiente Figura 3.18.

```
tesis@ chef-worstation:~/chef-repo$ ssh-keygen -b 4096
```

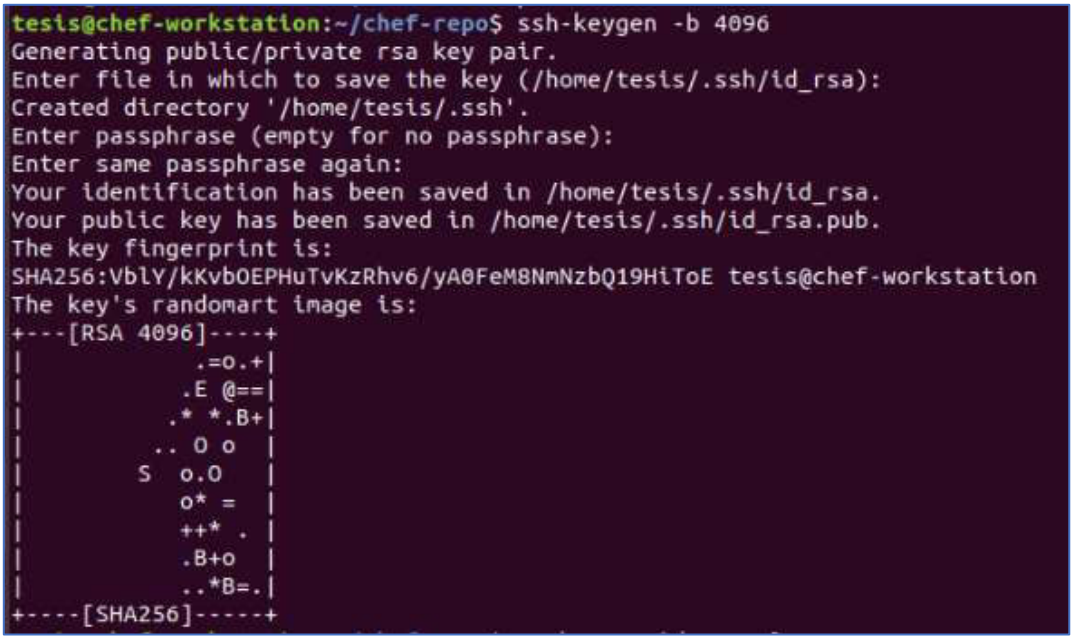


**Figura 3.18** Creación claves RSA

Se copiarán las claves que se generaron en el servidor Chef, hacia la estación de trabajo con el siguiente comando, esto se lo realiza para entablar una comunicación entre máquinas de confianza, se debe tomar en cuenta que se usará la dirección del servidor 192.168.427.140 está ya conocida anteriormente.

```
tesis@chef-worstation: ~/chef-repo$ sudo ssh-copy-id
tesis@192.168.247.209
```

Entablada la comunicación, el *software* solicitará la contraseña que pertenece al servidor de esta forma se comunicarán entre el servidor y la estación de trabajo con la siguiente línea de comandos, permitiendo copiar los archivos **.pem** como se presentarán a continuación en la Figura 3.19.

```
tesis@chef-worstation: ~/chef-repo$ scp tesis@192.168.247.209:
~/.chef/*.pem ~/chef-repo/.chef/
```


**Figura 3.19** Copiar archivos .pem.

Se prosigue a la comprobación correspondiente de los archivos **.pem** en la estación de trabajo.

```
tesis@chef-worstation:~/chef-repo$ ls ~/chef-repo/.chef
```

Adicionalmente se debe descargar ssh en el cliente y copiarse las claves generadas por ssh en la estación de trabajo, procedimiento que se mostrará en las configuraciones de Chef-*Client* (Nodo).

**Generar control de versiones**

Para el seguimiento de cambios en la realización de libros de cocina, mientras se efectúan tareas de configuración en la estación de trabajo, estos cambios y ediciones necesitan un control, este se lo realizará mediante Git, donde se proseguirá a la instalación de complementos git para poder realizar las distintas configuraciones del mismo.

```
tesis@chef-worstation:~$ sudo apt-get install git
```

Se empezará creando un repositorio git dentro de chef-repo, sucesivamente se agregará el nombre de usuario, así como correo electrónico para la configuración, esto se realizará mediante las siguientes líneas de código.

```
tesis@chef-worstation:~$ git config  --global  user.name jennyssale

tesis@chef-worstation:~$ git config  --global  user.email
jennifer_robalino1997@hotmail.com
```

Con la siguiente línea de código se agregará el directorio **.chef** al archivo **.gitignore**.

```
tesis@chef-worstation:~$ echo ".chef" > ~/chef-repo/.gitignore
```

Se ingresará al directorio chef-repo para agregar y confirmar los comandos, donde posteriormente se observará la creación de los distintos libros de configuración como se muestra en la Figura 3.20.

```
tesis@chef-worstation:~$ cd ~/chef-repo

tesis@chef-worstation:~/chef-repo$ sudo git init

tesis@chef-worstation:~/chef-repo$ sudo git add .

tesis@chef-worstation:~/chef-repo$ sudo git commit -m “initial
```

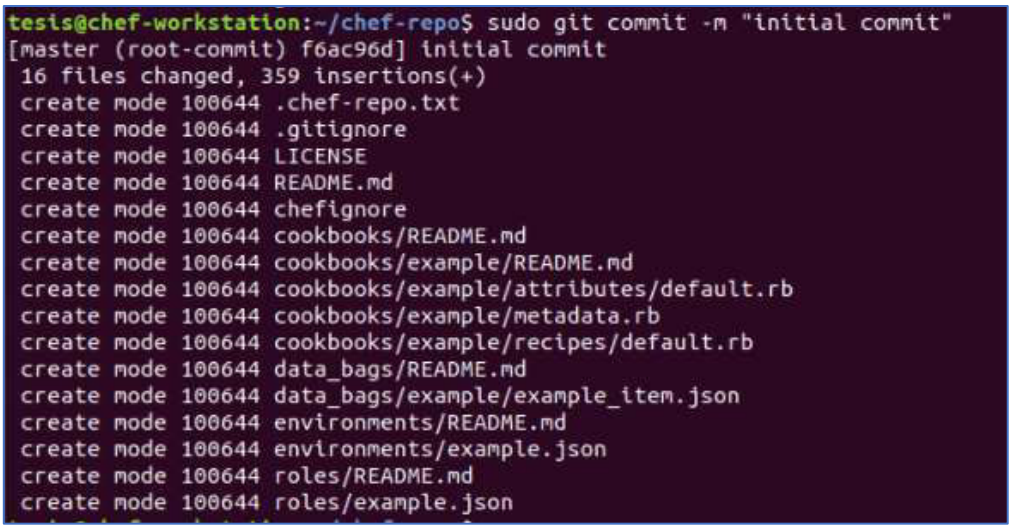


**Figura 3.20** Comandos git desde el directorio.

Se prosigue a ingresar el siguiente comando para la verificación del estado.

```
tesis@chef-worstation:~/chef-repo$ git status

On branch master

nothing to commit, working tree clean
```

**Instalación de Ruby**

Ruby es un lenguaje de programación requerido en Chef este ayudará con las configuraciones y especificaciones que se desee dar a un libro de cocina, por lo cual permitirá tener una sintaxis en el entorno en que se trabajará, el mismo se instalará en la *Workstation* y el nodo cliente.

Mediante la instalación posteriormente se podrá leer las recetas, que se encontrará más adelante, estas se podrán identificar con la extensión **.rb** , se prosigue a ingresar la siguiente línea de código.

```
tesis@chef-worstation:$ sudo apt-get install ruby-full
```

Para la verificación de la instalación de Ruby se ingresará el siguiente comando, posteriormente se obtendrá una salida como se presenta en la Figura 3.21.

```
tesis@chef-worstation:$ ruby --version
```


**Figura 3.21** Verificación de instalación de Ruby

Para poder elaborar una programación eficiente en Ruby, se requiere conocer los comandos de edición con vi o vim, información que se observa a mayor detalle en el **Anexo 2**.

### Configuración de *Knife*

El cuchillo es una de las herramientas que permitirá administrar distintos nodos, libros de cocina y recetas; para la correspondiente configuración del mismo se requerirá crear un archivo **config.rb**, este se realizará moviéndose al directorio de **.chef**.

```
tesis@chef-worstation:$ sudo nano ~/chef-repo/.chef/config.rb
```

Ingresado el comando se abrirá el archivo config.rb donde se editará los siguientes parámetros mostrados en la Figura 3.22, posteriormente se guardarán los cambios realizados.


**Figura 3.22** Archivo config.rb

Se prosigue a la copia de certificados SSL, los cuales se realizarán moviéndose al directorio **chef-repo**, ingresando las siguientes líneas de comando, donde se mostrará la ejecución de este como en la Figura 3.23.

```
tesis@chef-worstation:$ cd chef-repo

tesis@chef-worstation:/chef-repo$ knife ssl fetch
```


**Figura 3.23** Copia de certificados SSL

Se continúa verificando la ejecución correcta del archivo config.rb, ingresando el siguiente comando el cual mostrará el nombre del *validator*.

```
tesis@chef-worstation:~/chef-repo$ knife client list

chef-org-validator
```

### Bootstrap de un nodo

Posterior a la creación de la máquina Cliente en Ubuntu 18.04, y verificada la dirección con la que trabajará, se realizará mediante el comando Bootstrap el arranque del nodo, esta configuración se la realiza en la máquina de la estación de trabajo, para que posteriormente el servidor mantenga una comunicación con el nodo cliente el cual usará el usuario y contraseña ya establecidos anteriormente.

Se ejecutará el comando de arranque de cuchillo de cocina en conjunto de Bootstrap desde la *Workstation* y asegurarse estar dentro del directorio **chef-repo**, para cuando este comando se ejecute solicitará la clave del nodo y se presentará como se muestra en la siguiente Figura 3.24.

```
tesis@chef-worstation: ~/chef-repo$ knife bootstrap 192.168.247.211
-ssh-user tesis – sudo –identity-file ~/.ssh/id_rsa.pub --node-
name chefadmin
```

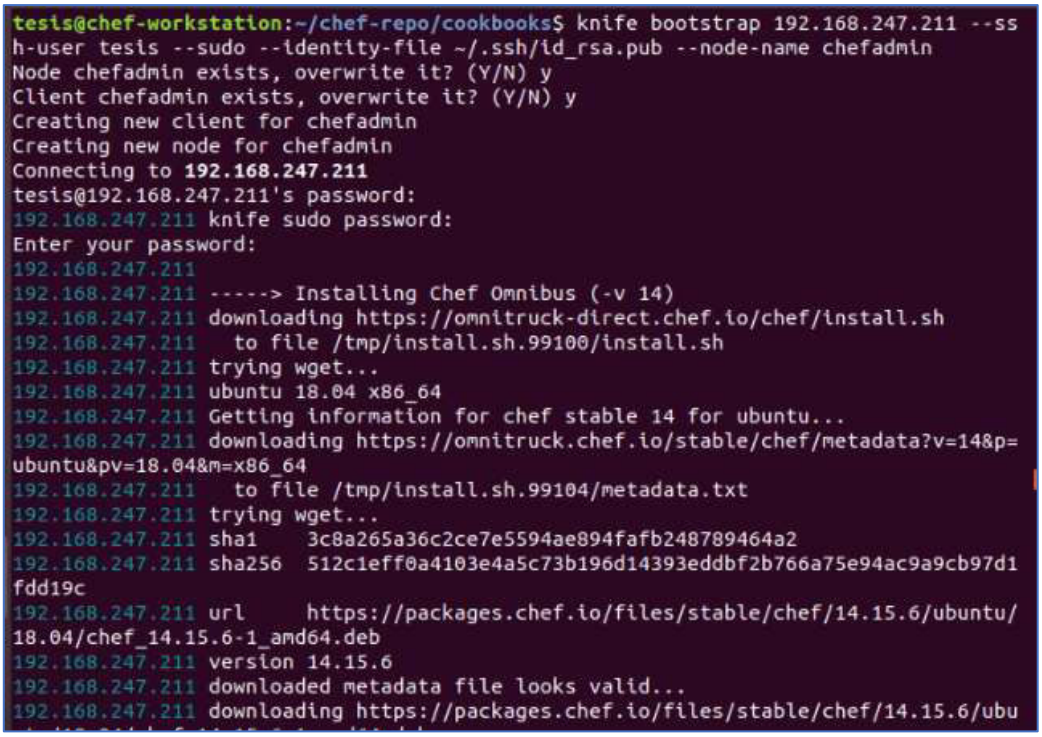


**Figura 3.24** Arranque de nodo

Posteriormente arrancado el nodo se verificará que el mismo se ha validado con las siguientes líneas de código como en la Figura 3.25, adicional, confirmar en la interfaz web del servidor que el nodo se encuentre registrado como se indica en la Figura 3.26.

```
tesis@chef-worstation: ~/chef-repo$ knife node list
tesis@chef-worstation: ~/chef-repo$ knife node show chefadmin
```


**Figura 3.25** Verificación de arranque de nodo en *Workstation*.

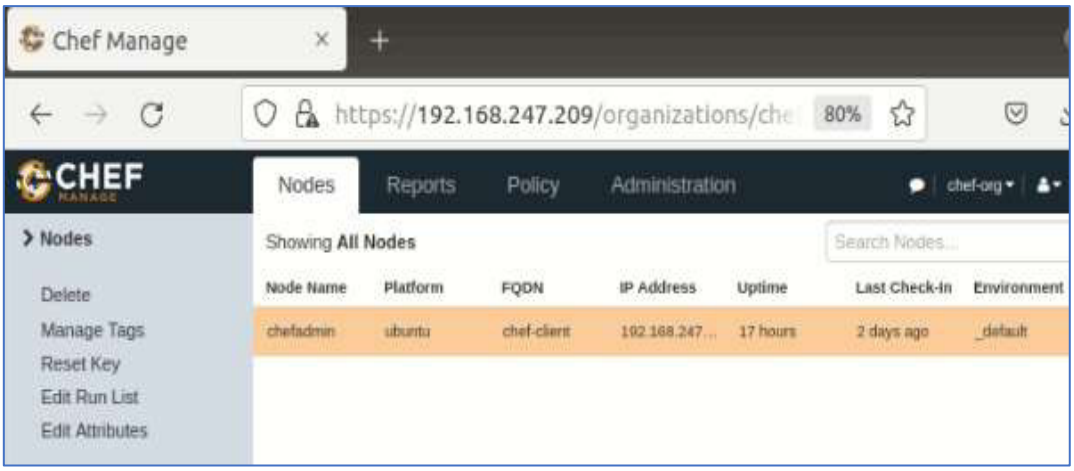


**Figura 3.26** Verificación de arranque de nodo en Chef *Manage*

### Creación libro de Prueba

Una vez completa la instalación de la estación de trabajo se creará un libro de cocina de prueba, para la verificación correcta del nodo y *cookbooks*. Cuando este sea subido al servidor se generará la compilación en el nodo; esta acción se procederá con el siguiente comando que creará un libro de cocina o también llamados en inglés *cookbooks*, esto se podrá visualizar en la Figura 3.27.

```
tesis@chef-worstation:~/chef-repo$ chef generate cookbook
cookbooks/libro_1
```

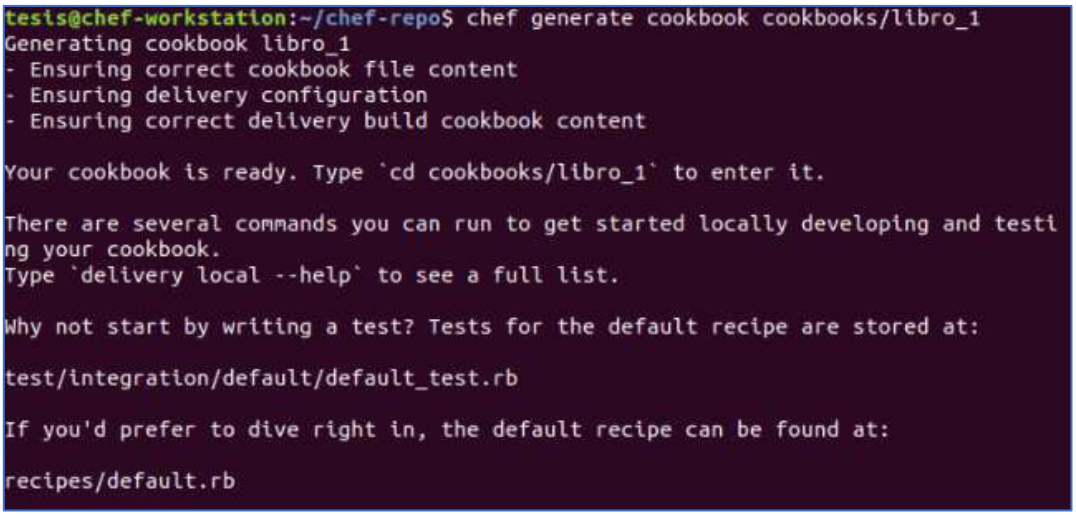


**Figura 3.27** *Cookbook* generado (libro_1)

Creado el libro, se tiene que editar el archivo **metadata.rb** donde se especificará el nombre y correo electrónico del usuario, ingresando el siguiente comando se ingresará al archivo y se escribirá lo siguiente como se indica en la Figura 3.28.

```
tesis@chef-worstation:~$ sudo vi chef-repo/cookbooks/libro_1/
metadata.rb
```


**Figura 3.28** Archivo de metadata.rb editado

A continuación, se editará la receta predeterminada por defecto, cuando se creó el libro de prueba, se ingresará el siguiente comando y se incorporará las siguientes líneas dentro del archivo **default.rb** como se muestra en la Figura 3.29.

```
tesis@chef-worstation:~$ sudo vi chef-repo/cookbooks/libro_1/
recipes/default.rb
```

```
#
# Cookbook:: libro_1
# Recipe:: default
#
# Copyright:: 2021, The Authors, All Rights Reserved.
hostname = node['hostname']
file '/etc/motd' do
  content "HOLA #{hostname}\n Este es el servidor Chef de JR\n\n"
end
```

**Figura 3.29** Archivo default.rb editado

A continuación, se subirá el libro de cocina de prueba al servidor por medio del comando especificado, y se presentará la creación del libro en la interfaz gráfica Chef *Manage* de la siguiente manera como se indica en la Figura 3.30, para posteriormente ejecutarlo en el nodo cliente.

```
tesis@chef-worstation:~/chef-repo/cookbooks$knife upload cookbook
libro_1
```

```
tesis@chef-workstation:~$ cd chef-repo/
tesis@chef-workstation:~/chef-repo$ cd cookbooks/
tesis@chef-workstation:~/chef-repo/cookbooks$ knife upload cookbook libro_1
Created libro_1
tesis@chef-workstation:~/chef-repo/cookbooks$
```

**Figura 3.30** Libro_1 subido al servidor

### Descarga libros de cocina de supermercado

Existen libros de cocina creados, donde Chef proporciona una interfaz para poder descargarlos y ejecutarlos estos se encontrarán en Chef Supermarket, este repositorio de libros de cocina es realizados por la comunidad.

Subsiguiente se tiene que descargar con las siguientes líneas de código los recetarios chef-*client*, cron y logrotate estos se encontrarán en el repositorio de supermercado, lo descrito anteriormente se puede ver en la Figura 3.31.

```
$ knife supermarket download chef-client
$ knife supermarket download cron
$ knife supermarket download logrotate
```

```
tesis@chef-workstation:~/chef-repo/cookbooks$ knife supermarket download chef-clie
nt
Downloading chef-client from Supermarket at version 12.3.4 to /home/tesis/chef-rep
o/cookbooks/chef-client-12.3.4.tar.gz
Cookbook saved: /home/tesis/chef-repo/cookbooks/chef-client-12.3.4.tar.gz
tesis@chef-workstation:~/chef-repo/cookbooks$ knife supermarket download cron
Downloading cron from Supermarket at version 7.0.2 to /home/tesis/chef-repo/cookbo
oks/cron-7.0.2.tar.gz
Cookbook saved: /home/tesis/chef-repo/cookbooks/cron-7.0.2.tar.gz
tesis@chef-workstation:~/chef-repo/cookbooks$ knife supermarket download logrotate
Downloading logrotate from Supermarket at version 3.0.3 to /home/tesis/chef-repo/c
ookbooks/logrotate-3.0.3.tar.gz
Cookbook saved: /home/tesis/chef-repo/cookbooks/logrotate-3.0.3.tar.gz
```

**Figura 3.31** Descarga *cookbooks* de supermercado

Posteriormente se observa que las descargas se realizan en archivos **.tar.gz** , para seguir con la descompresión de cada uno de ellos, estos deberán descomprimirse dentro del repositorio de Chef local de *cookbooks*.

```
$ tar -xvzf chef-client-12.3.4.tar.gz
$ tar -xvzf cron-7.0.2.tar.gz cron
$ tar -xvzf logrotate-3.0.3.tar.gz
```

Descomprimidos los libros se verifica que se encuentren ahora en su repositorio local, para reanudar con la subida de los libros al servidor, se debe verificar que estos se encuentren en el historial de libros con sus respectivas versiones en Chef Manage como se indica en la Figura 3.32.

```
$ knife upload cookbook chef-client
$ knife upload cookbook cron
$ knife upload cookbook logrotate
```

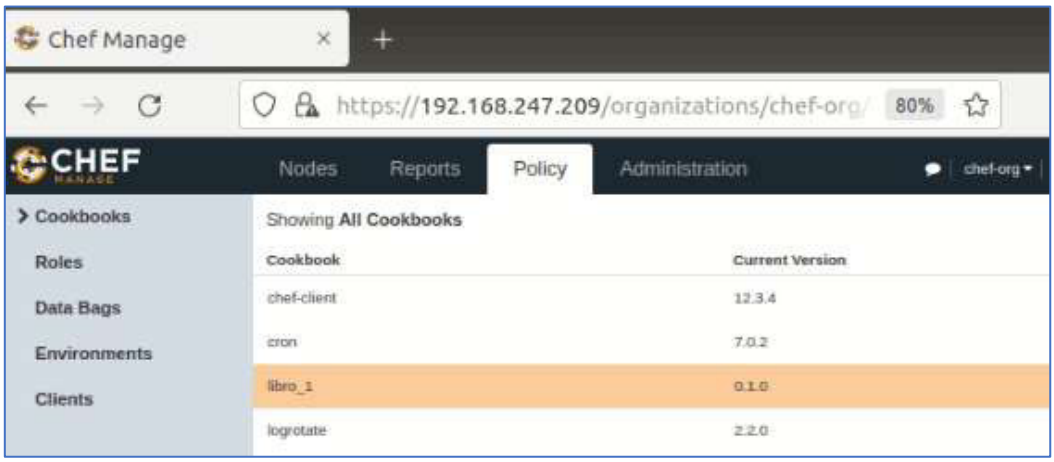


**Figura 3.32** Libros de cocina subidos al servidor

En el **Anexo 3** se muestra el listado de todos los comandos ingresados en la estación de trabajo de Chef, que fueron previamente explicados.

### Instalación y configuración Chef-Client (Chef Infra Client)

Chef -Client es aquel que se ejecutará localmente en los nodos que serán administrados por Chef Infra *Server*, de esta forma la máquina cliente entablará una comunicación con el servidor para dotar a la máquina de las configuraciones que se requieran en dicho nodo para llevarlo al estado esperado.

Para entablar una comunicación segura mediante las máquinas de la infraestructura se creará el repositorio **.chef** , donde se guardarán las claves con las cuales se trabajará.

```
tesis@chef-client:~$ mkdir .chef
tesis@chef-client:~$ cd .chef
tesis@chef-client:~ /.chef$
```

A continuación, se copiarán las claves generadas en la *Workstation* hacia el nuevo directorio creado con el siguiente comando, en este se podrá verificar los archivos .**pem** copiados como se muestra en la Figura 3.33.

```
tesis@chef-client:~/.chef$ scp tesis@192.168.247.209:~/.chef/*.pem
~/.chef/
```

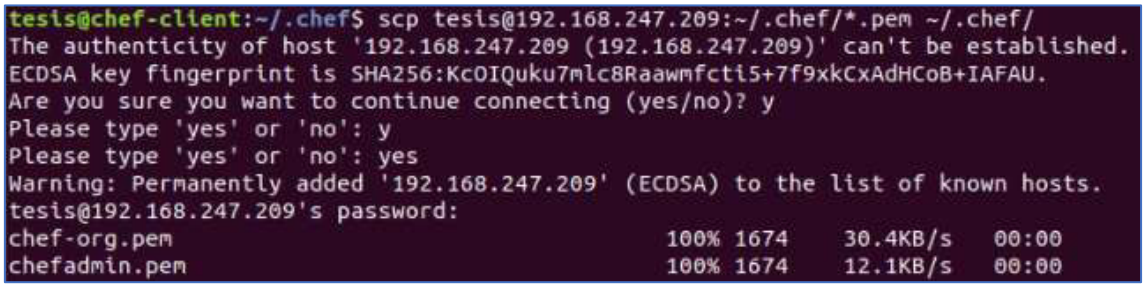


**Figura 3.33** Copia de archivos .pem en repositorio .chef

Se podrá visualizar que se entablará una comunicación con el servidor con la siguiente línea de código y se podrá visualizar como se presenta en la Figura 3.34.

```
tesis@chef-client:~$ ssh 192.168.247.209
```

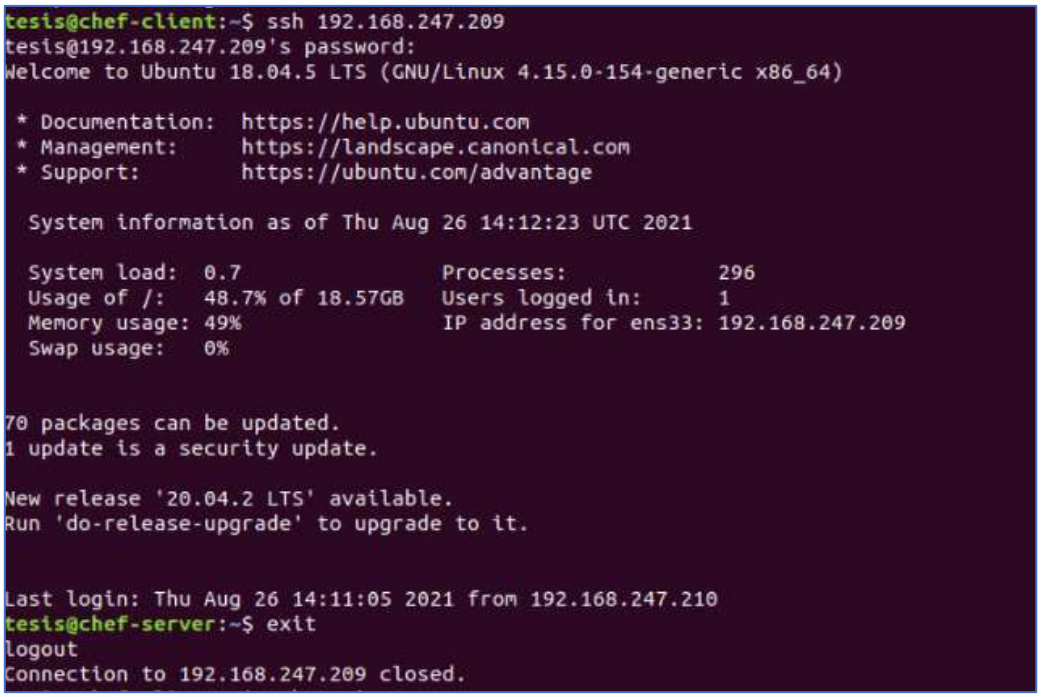


**Figura 3.34** Comunicación mediante claves .pem con el servidor .

Se prosigue a la instalación de OpenSSH mediante la siguiente línea de código, para poder entablar comunicación con la estación de trabajo, de esta manera se podrá ejecutar los libros de cocina desde la estación de trabajo si fuese el caso, con la instalación de la misma se establece una comunicación segura para la comunicación remota de los equipos.

Se necesita descargar OpenSSH *Server* para tener una conexión remota desde la estación de trabajo hacia el nodo, esto se realiza con los siguientes comandos.

```
tesis@chef-client:~$ sudo apt-get install openssh-server
tesis@chef-client:~$ sudo systemctl start sshd.service
tesis@chef-client:~$ sudo systemctl stop sshd.service
tesis@chef-client:~$ sudo systemctl restart sshd.service
```

Una vez realizados estos cambios se verifica que el servicio esté activo, como se visualiza en la Figura 3.35 y así mismo se entable comunicación con la estación de trabajo como se indica en la Figura 3.36.

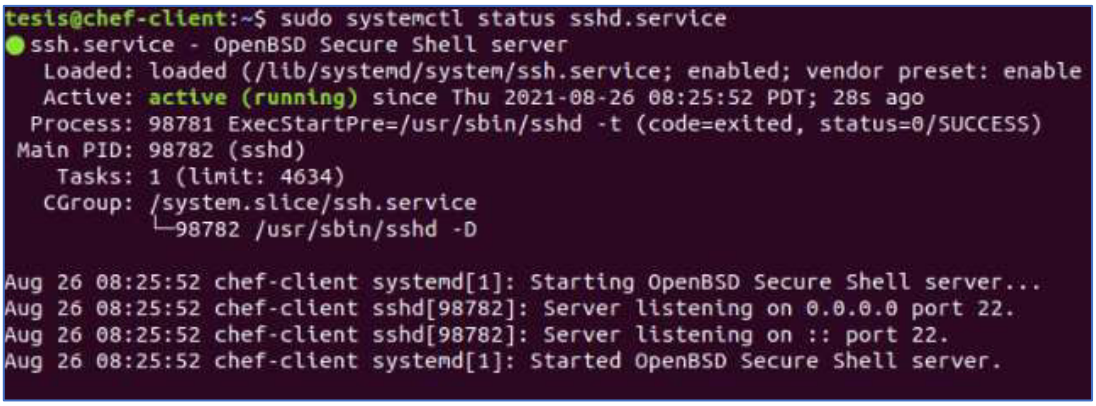


**Figura 3.35** Verificación servicio activo SSH

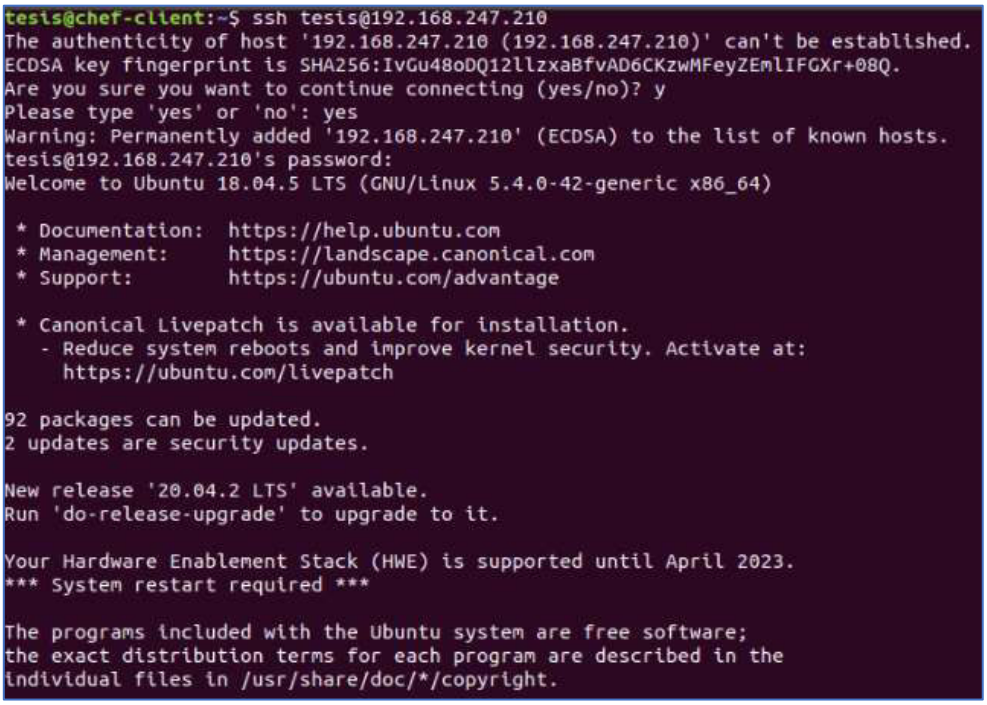


**Figura 3.36** Comunicación ssh con la *Workstation*

**Ejecución de libros de cocina en Chef-Client**

Comprobado los libros de cocina disponibles en el servidor, se ejecuta los mismos *cookboks* mediante Chef Manage, en el cual se seleccionará el nodo en el que se desea trabajar, para posteriormente editar la lista de ejecución.

Se ingresa a Chef Manage y dirigirse al nodo, ya en los nodos, en la pestaña de acciones seleccionar la opción *Edit Run List*, estas opciones se observan en la Figura 3.37.

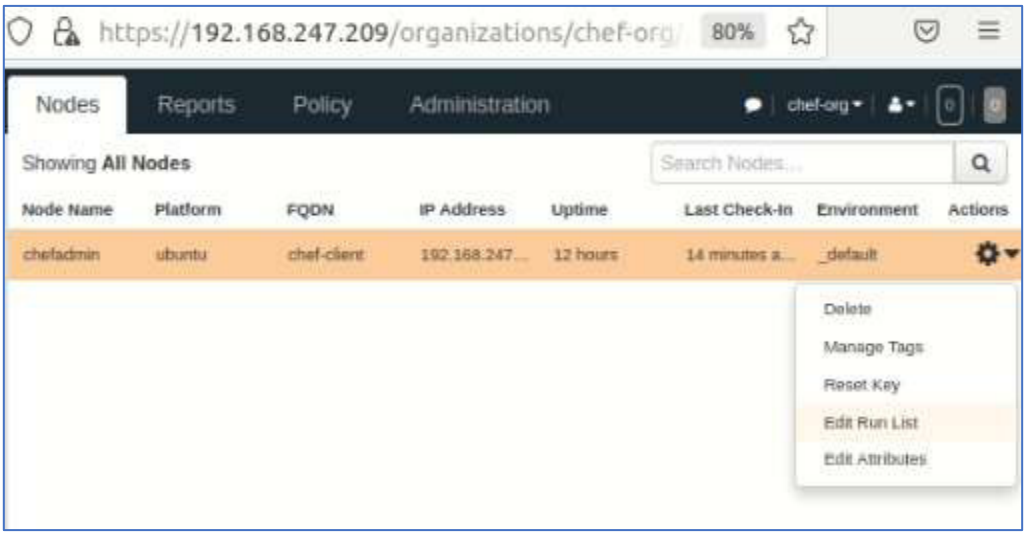


**Figura 3.37** Editar lista de ejecución

Abierta la lista de ejecución se seleccionan las recetas para la ejecución de su nodo, estas podrán ser deslizadas desde recetas disponibles hasta la lista de ejecución actual. Para el siguiente caso se ejecutarán las recetas del libro de prueba y **chef-client**, al ejecutarse las recetas, harán uso de extensiones de los libros como **cron** y **logrotate**, seguido de esto se guarda los cambios, estos pasos se podrán visualizar en la Figura 3.38.

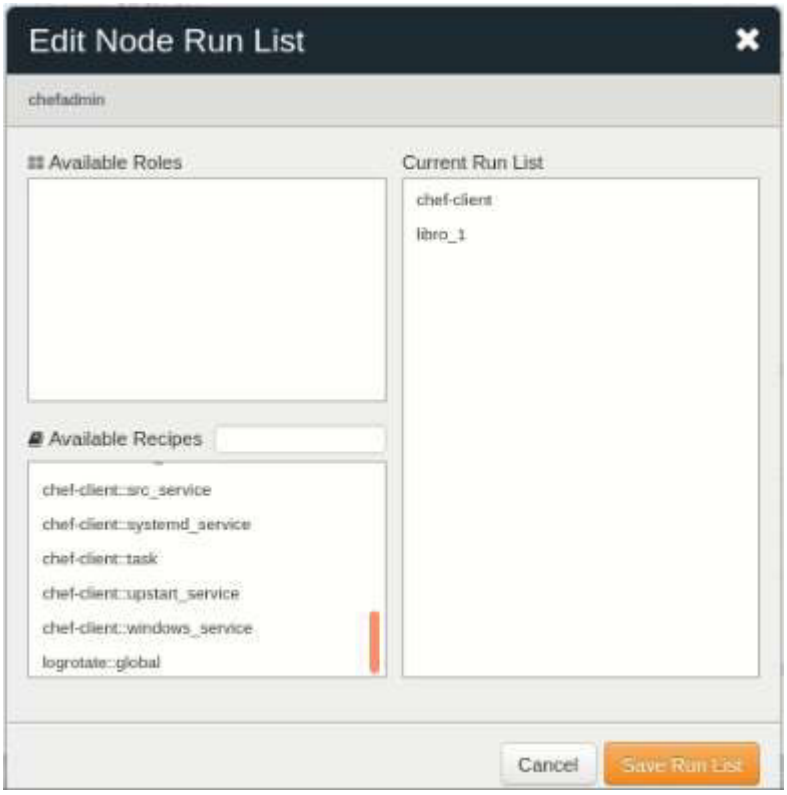


**Figura 3.38** Lista de ejecución actual

Siguiente a los cambios realizados en la máquina del nodo cliente, ingresar el siguiente comando como se observa en la Figura 3.39 y Figura 3.40 la ejecución de las recetas, esta comprobación también se la puede realizar desde la estación de trabajo ingresando por medio de ssh al cliente.

```
tesis@chef-client:~$ sudo Chef-client
```

```
tesis@chef-client:~$ sudo chef-client
[sudo] password for tesis:
Starting Chef Client, version 14.15.6
resolving cookbooks for run list: ["chef-client", "libro_1"]
Synchronizing Cookbooks:
  - cron (7.0.2)
  - chef-client (12.3.4)
  - libro_1 (0.1.0)
  - logrotate (2.2.0)
Installing Cookbook Gems:
Compiling Cookbooks...
[2021-08-26T09:32:17-07:00] WARN: Resource cron_access from the client is overridi
ng the resource from a cookbook. Please upgrade your cookbook or remove the cookbo
ok from your run_list.
[2021-08-26T09:32:17-07:00] WARN: Resource cron_d from the client is overriding th
e resource from a cookbook. Please upgrade your cookbook or remove the cookbook fr
om your run_list.
Converging 10 resources
Recipe: chef-client::systemd_service
  * directory[/var/run/chef] action create
    - create new directory /var/run/chef
    - change owner from '' to 'root'
    - change group from '' to 'root'
  * directory[/var/lib/chef] action create
    - create new directory /var/lib/chef
    - change owner from '' to 'root'
    - change group from '' to 'root'
  * directory[/var/log/chef] action create
    - create new directory /var/log/chef
```

**Figura 3.39** Ejecución de libro de cocina chef-client en el nodo

```
    - change owner from '' to 'root'
    - change group from '' to 'root'
    - creating unit: chef-client.service
Recipe: chef-client::systemd_service
  * systemd_unit[chef-client.timer] action stop (up to date)
  * systemd_unit[chef-client.timer] action disable (up to date)
  * systemd_unit[chef-client.timer] action delete (up to date)
  * service[chef-client] action enable
    - enable service service[chef-client]
  * service[chef-client] action start
    - start service service[chef-client]
Recipe: libro_1::default
  * file[/etc/motd] action create
    - create new file /etc/motd
    - update content in file /etc/motd from none to 89f479
    --- /etc/motd       2021-08-26 09:32:19.601004234 -0700
    +++ /etc/.chef-motd20210826-100910-1kpcmu2  2021-08-26 09:32:19.601004234 -070
0
    @@ -1 +1,4 @@
    +HOLA chef-client
    + Este es el servidor Chef de JR
    +
Recipe: chef-client::systemd_service
  * service[chef-client] action restart
    - restart service service[chef-client]

Running handlers:
Running handlers complete
Chef Client finished, 10/15 resources updated in 08 seconds
```

**Figura 3.40** Ejecución de libro de cocina libro_1 en el nodo

Se comprobará la ejecución del *cookbook* libro_1, desde la estación de trabajo, donde se ingresará el siguiente comando, en este se podrá visualizar en el banner SSH del nodo lo que se muestra en la Figura 3.41, que son las líneas de texto ingresadas en la receta **default.rb** anteriormente editadas.

```
tesis@chef-workstation:~$ ssh tesis@192.168.247.211
```

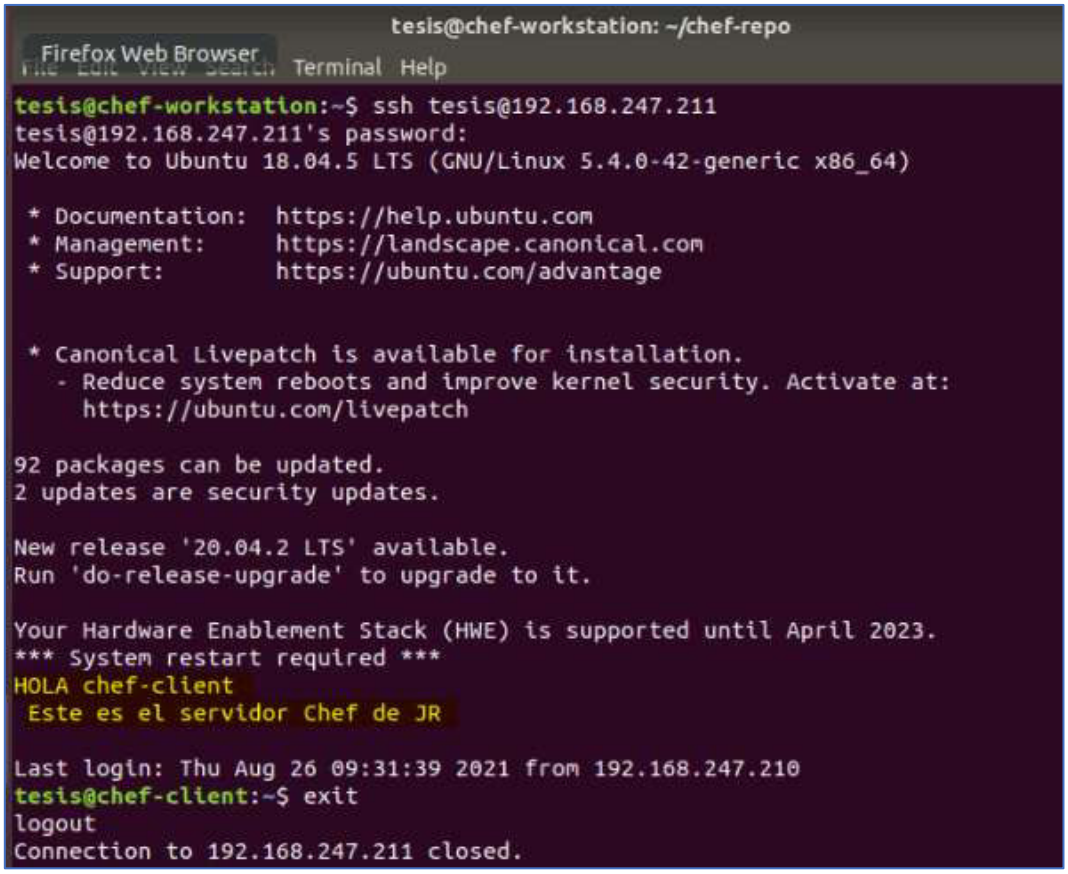


**Figura 3.41** Comprobación libro_1

En el **Anexo 4** se mostra el listado de todos los comandos ingresados en el nodo cliente, que fueron previamente explicados.

## 3.3 Simulación de los diferentes entornos que se pueden presentar en un ambiente a implementar mediante Chef

### Despliegue del servidor DNS

Para la instalación del servidor Sistema de nombres de dominio (DNS), se hace uso del paquete Bind9.La creación de este servidor permitirá tener un identificador para cada dirección IP que está asociada al dispositivo conectado a la red, permitiendo enlazar el nombre con la dirección IP, facilitando el transporte del contenido que esté alojado en la dirección asociada al dominio [20].

Para la configuración de Bind9 se creará un libro de cocina en la estación de trabajo, el mismo que tendrá la receta de instalación y configuración del servidor DNS, mediante la siguiente línea de comando se establecerá el nuevo el *cookbook*. Como se visualiza en la Figura 3.42 ,ejecutado el comando se verán recetas, valores de atributos, archivos y plantillas que por defecto poseen los *cookbooks* para definir el escenario y contenido a trabajar.

Desde el archivo raíz se tiene que dirigir al directorio **chef-repo**/***cookbook*s** donde se crea el libro de cocina titulado bind9, seguidamente se verá la estructura creada mediante el comando de árbol, esto se visualiza en la **Figura *3.1***Figura 3.43.

```
tesis@chef-workstation:~$ cd chef-repo/cookbooks
tesis@chef-workstation:~/chef-repo/cookbooks$ chef generate
cookbook bind9
```

```
tesis@chef-workstation:~/chef-repo/cookbooks$ chef generate cookbook bind9
Generating cookbook bind9
- Ensuring correct cookbook file content
- Ensuring delivery configuration
- Ensuring correct delivery build cookbook content

Your cookbook is ready. Type `cd bind9` to enter it.

There are several commands you can run to get started locally developing and testin
g your cookbook.
Type `delivery local --help` to see a full list.

Why not start by writing a test? Tests for the default recipe are stored at:

test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:

recipes/default.rb
```

**Figura 3.42** Creación de libro de cocina bind9

```
tesis@chef-workstation:~/chef-repo/cookbooks$ cd bind9/
tesis@chef-workstation:~/chef-repo/cookbooks/bind9$ tree
```

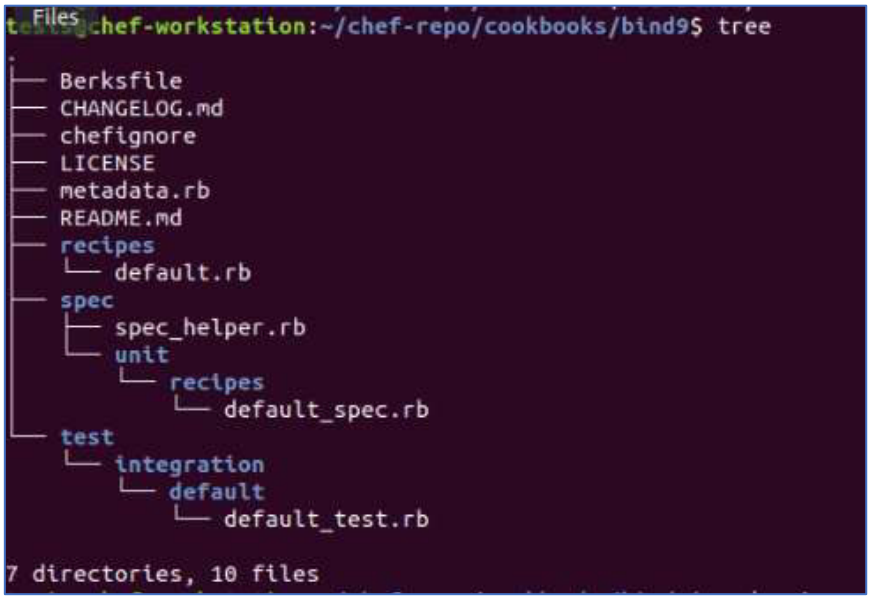


**Figura 3.43** Estructura inicial de libro de cocina bind9

Una vez creado el libro se instalará el paquete de bind9. Se configurará y se pondrá en marcha, esto se realizará ingresando a *recipes* que es un directorio predefinido de *cookbook*, este contiene el archivo **default.rb** que es una pieza de datos que ayudará a chef-client a determinar el estado actual del nodo o cualquier cambio que se realice y se quiera poner en ejecución en el nodo. Este es archivo es el que se cargará primero, siguiendo las siguientes líneas de comando se ingresará al archivo **default.rb**.

```
tesis@chef-workstation:~/chef-repo/cookbooks/bind9$ sudo vi
recipes/default.rb
```

En la Figura 3.44 se observará por defecto el comentario del nombre del *cookbook* y la receta, de esta forma se puede identificar que el archivo que se modificará es el correcto. Este archivo acepta escritura en el lenguaje de programación Ruby, se define el recurso del paquete para instalar, seguido se llama al paquete bind9 en el que se define la habilitación y el inicio del servicio DNS. En este archivo se agregarán las siguientes líneas de código, se guardará y cerrará el archivo **default.rb**.

```
package "bind9" do
  action :install
end

service "bind9" do
  supports :status => true, :reload => true, :restart => true
  action [:enable, :start]
end
```


**Figura 3.44** Instalación, habilitación y reinició de servicio bind9

Bind9 posee el archivo de configuración específica del servidor DNS, que se encuentra en /**etc/bind/named.conf.local**, en este archivo se realizará las configuraciones del DNS maestro, donde se determinará un nombre a todos los equipos de la red. Para el cual se instalará un servidor DNS privado con un dominio ficticio para el caso de simulación. Caso contrario, de ser una implementación real, se añadirá el dominio correspondiente a la organización; todas las máquinas pertenecientes a la infraestructura creada corresponden a dicho dominio, trabajando en la red interna [21].

El servidor DNS maestro instalado en el nodo será apto de resolver las peticiones internas de nombres de dominio, de forma directa como inversa, es decir, si se ingresa una consulta con una dirección IP, se verá responder y resolver el nombre de dominio asignado a cada dirección y viceversa. Es por consiguiente que en el archivo **named.conf.local** se verá detallado la especificación de dominio de la zona inversa y directa [21].

Dentro del directorio **chef-repo**, se tendrá que ejecutar el comando de generación de *templates* con el nombre de la plantilla definida, donde la función de plantilla es necesaria para poder crear los archivos de host virtuales necesarios para la instalación

de dicho servicio en el nodo. Se ejecutará la siguiente línea de código para la creación de la plantilla **named.conf.local.rb** la misma después de ser ejecutada se verá como se muestra en la Figura 3.45.

```
:~/chef-repo/cookbooks/bind9$ chef generate template
named.conf.local
```


**Figura 3.45** Creación de *template* named.conf.local.

Sucesivamente creado el *templete*, se tiene que abrir y editar el archivo, usando variables de Ruby, este archivo se identificará como una plantilla al tener una extensión **.erb** , los nombres de las variables creadas serán definidas en el archivo de receta para el momento de la llamada de ejecución.

Se escribe el siguiente comando para ingresar al archivo, donde posteriormente abierto el archivo, se editará el contenido como se indica en la Figura 3.46.

```
$ sudo vi chef-repo/cookbooks/bind9/templates/named.conf.local.erb
```

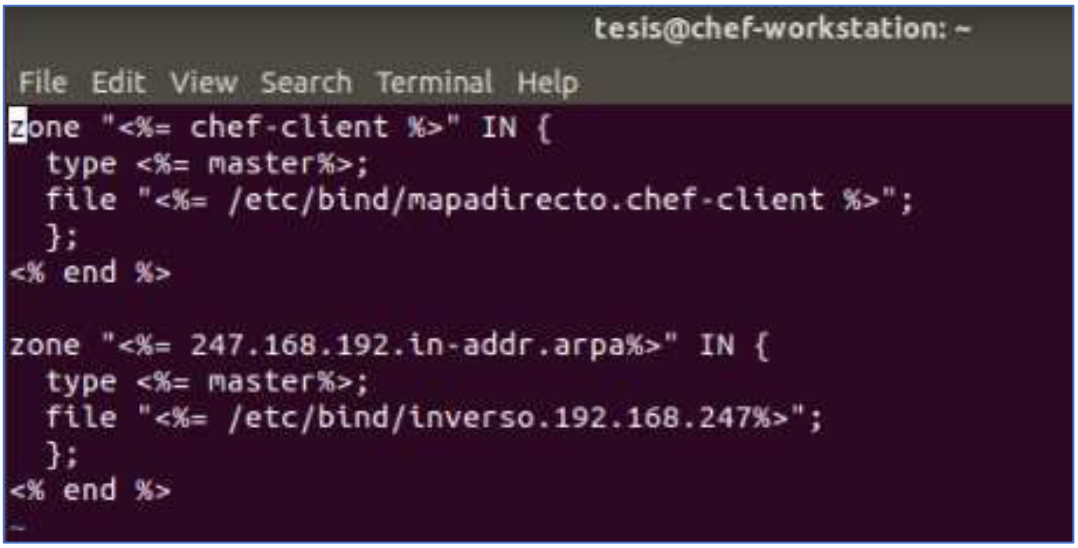


**Figura 3.46** Configuraciones *template* named.conf.local.erb

Una vez editado, guardado y cerrado el archivo, se prosigue a la edición de la receta **default.rb.** Detrás de los comandos de instalación se agregarán líneas de código en Ruby llamando a la plantilla anteriormente creada y editada. El nombre del *template* debe ser la ubicación del host virtual en el nodo, la fuente será el nombre del archivo del *template* creado, se integrará el modo, que es el que concederá al propietario privilegios, estos pueden ser de lectura, escritura y el resto de los privilegios de lectura como se puede ver en la Figura 3.47.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/bind9/
recipes/default.rb
```

```
template "/etc/bind/named.conf.local" do
source "named.conf.local.erb"
  owner "root"
  group "root"
  mode 0644
end
```

```
#
# Cookbook:: bind9
# Recipe:: default
#
# Copyright:: 2021, The Authors, All Rights Reserved.

package "bind9" do
  action :install
end

service "bind9" do
  supports :status => true, :reload => true, :restart => true
  action [:enable, :start]
end

template "/etc/bind/named.conf.local" do
source "named.conf.local.erb"
  owner "root"
  group "root"
  mode 0644
end
```

**Figura 3.47** Definición de variables de ejecución de *template* named.conf.local.erb

Acto seguido, se realizará la configuración del archivo de zona de búsqueda directa, en la cual nuevamente se creará el archivo de plantilla que tendrá el nombre de **mapadirecto.chef-client**. La ubicación de este libro se especificó al momento de crear los parámetros descriptivos de maestro para el dominio, en la plantilla

**named.conf.local**, en la Figura 3.48 se podrá observar la creación del *template* dentro del directorio *cookbook* bind9 mediante la siguiente línea de código.

```
:~/chef-repo/cookbooks/bind9$ chef generate template
mapadirecto.chef-client
```


**Figura 3.48** Creación de *template* mapadirecto.chef-*client*.

Creada la plantilla, se ingresará al archivo **mapadirecto.chef-client.erb** *,* en este se configurará con entradas separadas donde se podrá ver registro de recursos divididos en nombre, TTL (tiempo de vida), clase, tipo y datos, como se indica en la Figura 3.49. Las primeras líneas editadas son parámetros afines con la actualización del DNS, donde se localizarán números de serie y etapas de actualización, seguido de indicar cual es el servidor primario (NS: *Name Server*), continuando con las líneas de especificación de las direcciones IP de las distintas máquinas que son componentes del dominio (A: *Address*).

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/bind9/
template/mapadirecto.chef-client.erb
```


**Figura 3.49** Configuración y edición de archivo de zona de búsqueda directa.

Configurada, guardada y cerrada la plantilla de búsqueda directa, se dará paso al llamado de ejecución en el archivo **default.rb**, se ingresará al mismo, en el cual se agregará los parámetros a continuación de los anteriormente descritos.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/bind9/
recipes/default.rb
```

```
template "/etc/bind/mapadirecto.chef-client" do
source "mapadirecto.chef-client.erb"
  owner "root"
  group "root"
  mode 0644
end
```

Tras haber guardado los cambios, se da paso a la creación de un nuevo *template* este llevará la información de la zona de búsqueda inversa, este ayudará a realizar búsquedas desde la dirección IP al nombre de dominio. Se hace uso nuevamente del comando de Chef para generar plantillas como se muestra a continuación en la Figura 3.50 y en la siguiente línea de código.

```
:~/chef-repo/cookbooks/bind9$ chef generate template
inverso.192.168.247
```

```
tesis@chef-workstation:~/chef-repo/cookbooks/bind9$ chef generate template inverso.
192.168.247
Recipe: code_generator::template
  * directory[/home/tesis/chef-repo/cookbooks/bind9/templates] action create (up to
 date)
  * template[/home/tesis/chef-repo/cookbooks/bind9/templates/inverso.192.168.247.er
b] action create
    - create new file /home/tesis/chef-repo/cookbooks/bind9/templates/inverso.192.1
68.247.erb
    - update content in file /home/tesis/chef-repo/cookbooks/bind9/templates/invers
o.192.168.247.erb from none to e3b0c4
    (diff output suppressed by config)
```

**Figura 3.50** Creación de *template* inverso.192.168.247.

A continuación, se editará el archivo **inverso.192.168.247.erb**, cómo se presenta en la Figura 3.51 se mantienen las configuraciones iniciales de números de serie y etapas de actualización, así como también el parámetro del servidor primario. Para este archivo se editará el registro de recurso con las siglas PTR, para enlazar las direcciones IP con los nombres de dominio, seguido se ingresará el último número de la dirección IP que apuntará al nombre de dominio completo o asociado que se encuentran descritos en la cuarta columna.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/bind9/
template/inverso.192.168.247.erb
```

```
$TTL <%= 604800%>
@ IN SOA <%=server.chef-client.%> <%=root.chef-client.%> (
                2       ; serial
                604800  ; refresh
                 86400  ; retry
                2419200  ; expiry
                 604800) ; Negative Cache TTL
;
                IN              NS              <%= server.chef-client. %>
<% 211%>
                IN              PTR              <%= server.chef-client.%>
<% 218%>
                IN              PTR              <%= client.chef-client. %>
<% 2%>
                IN              PTR              <%= router.chef-client.%>
<% 211%>
                IN              PTR              <%= correo.chef-client.%>

<% end %>
~
```

**Figura 3.51** Configuración y edición de archivo de zona de búsqueda inversa

Detrás de la edición de la plantilla se guarda y se cierra, para abrir nuevamente el archivo **default.rb**, donde se ingresará las líneas de código en lenguaje Ruby, para poder llamar a la plantilla anteriormente creada, como se indica a continuación.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/bind9/
recipes/default.rb
```

```
template "/etc/bind/mapadirecto.chef-client" do
source "mapadirecto.chef-client.erb"
  owner "root"
  group "root"
  mode 0644
end
```

Una vez configurados todos los parámetros descritos en el *cookbook* bind9, se verifica la creación y estructura de este mediante una lista de forma recursiva de los directorios y archivos. En la Figura 3.52 se podrá ver el directorio y archivos *template* creados anteriormente, sucesivamente se subirá el *cookbook* al servidor con ayuda de la herramienta *knife*. Mediante la siguiente línea de código; como se muestra en la Figura 3.53 al subir el libro se presentará el mensaje de libro de cocina subido con la versión del *cookbook*.

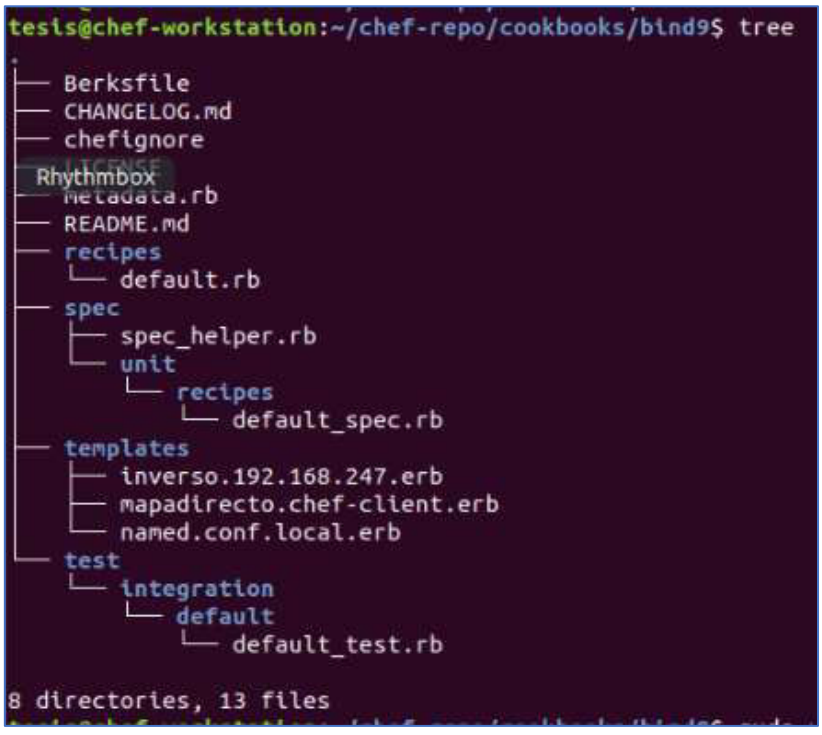


**Figura 3.52** Estructura final de libro de cocina bind9

tesis@chef-*workstation*:~/chef-repo/*cookbook*s$ knife *cookbook* upload bind9


**Figura 3.53** *Cookbook* bind9 subido al servidor.

Finalmente, se ingresa a la interfaz web de Chef Manage, en la opción de nodo, se editará la lista de ejecución como se muestra en la Figura 3.54, en esta parte se busca en recetas disponibles la receta bind9 para trasladarla a la lista de ejecución actual, se guarda los cambios, estos cambios se podrán visualizar cuando se ingrese en el nodo cliente.

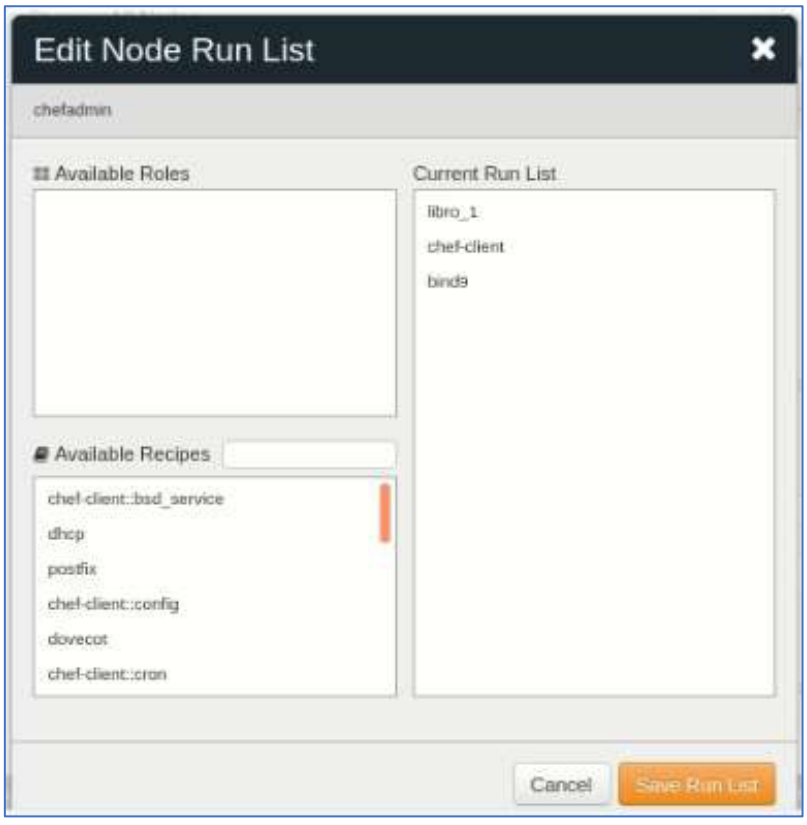


**Figura 3.54** Ejecución de *cookbook* bind9

### Despliegue del servidor DHCP

Para la instalación del servidor DHCP (Protocolo de configuración dinámica de host), se hace uso del paquete isc-dhcp-*server*. Este paquete contiene agentes de retransmisión y clientes, admitiendo comunicación IPv4. En este despliegue se requerirá la creación de dos adaptadores de red nuevos en el nodo cliente, esto se lo realiza con el objetivo de que no exista conflicto con la instalación de otros servidores propuestos y que el servidor DHCP trabaje en un adaptador de red con dirección única, así mismo el segundo adaptador de red creado será usado como cliente para la comprobación del servidor [22].

VMware brinda la opción de crear adaptadores, para esto se debe localizar la máquina a la que se quiere establecer dichas acciones como lo es Chef-*Client*, como se muestra en la Figura 3.55 dando clic izquierdo se desplegará una lista de parámetros, en las que se seleccionará la opción *Settings,* la cual abrirá una pestaña de más opciones de configuración.

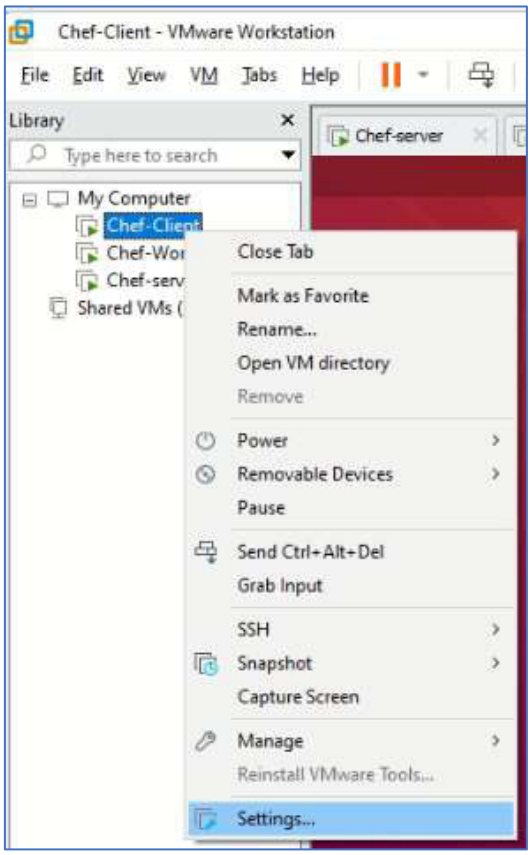


**Figura 3.55** Acceso a configuración de máquina virtual

Secuentemente en la ventana de configuración se dará clic en Agregar donde se abrirá una nueva ventana para añadir asistentes de *hardware*, para este caso se seleccionará

la opción *Network Adapter* y Finalizar como se presenta en la Figura 3.56. Por consiguiente, se verá la creación del adaptador, en el cual se dejará por defecto sus configuraciones en conexión NAT utilizado para compartir la dirección IP del host, finalmente se dará clic en OK para establecer todos los cambios.

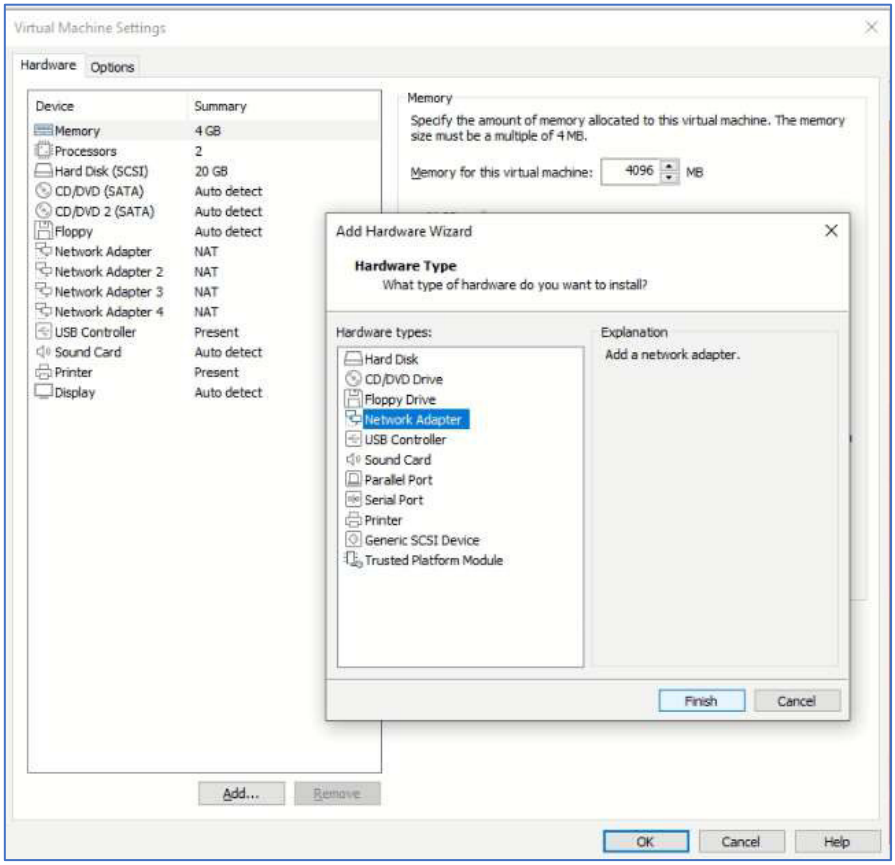


**Figura 3.56** Creación de nueva interfaz de red.

Creado el primer adaptador se repetirán los pasos para la creación del segundo adaptador de red, el cual trabajará como cliente para el servidor DHCP.

Para la creación de DHCP se creará un libro de cocina en la *Workstation*, el mismo que tendrá la receta de instalación y configuración del servidor DHCP. Mediante la siguiente línea de comando se establecerá el nuevo el *cookbook*, como se visualiza en la Figura 3.57. El mismo que contendrá recetas, valores de atributos, archivos y plantillas por defecto, dirigirse al directorio **chef-repo/*cookbook*s** donde creará el libro de cocina titulado dhcp, seguidamente se verá la estructura creada mediante el comando de árbol, esto se puede observar en la Figura 3.58.

```
tesis@chef-workstation:~$ cd chef-repo/cookbooks
tesis@chef-workstation:~/chef-repo/cookbooks$ chef generate
cookbook dhcp
```

```
tesis@chef-workstation:~/chef-repo/cookbooks$ chef generate cookbook dhcp
Generating cookbook dhcp
- Ensuring correct cookbook file content
- Ensuring delivery configuration
- Ensuring correct delivery build cookbook content

Your cookbook is ready. Type `cd dhcp` to enter it.

There are several commands you can run to get started locally developing a
nd testing your cookbook.
Type `delivery local --help` to see a full list.

Why not start by writing a test? Tests for the default recipe are stored a
t:

test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:

recipes/default.rb
```

**Figura 3.57** Creación de libro de cocina dhcp

```
tesis@chef-workstation:~/chef-repo/cookbooks$ cd dhcp/
tesis@chef-workstation:~/chef-repo/cookbooks/dhcp$ tree
```

```
tesis@chef-workstation:~/chef-repo/cookbooks/dhcp$ tree
.
├── Berksfile
├── CHANGELOG.md
├── chefignore
├── LICENSE
├── metadata.rb
├── README.md
├── recipes
│   └── default.rb
├── spec
│   ├── spec_helper.rb
│   └── unit
│       └── recipes
│           └── default_spec.rb
└── test
    └── integration
        └── default
            └── default_test.rb

7 directories, 10 files
```

**Figura 3.58** Estructura inicial de libro de cocina dhcp

Ingresando al directorio de *recipes* el cual contiene el archivo **defaul.rb**, se instalará el paquete de isc-dhcp-*server* adicional a esto se configurará y pondrá en marcha el servicio, se ingresará al archivo **default.rb** con las siguientes líneas de comando.

```
tesis@chef-workstation:~/chef-repo/cookbooks/dhcp$ sudo vi
recipes/default.rb
```

En la Figura 3.59 se observará que se define el recurso del paquete para instalar, seguido se llama al paquete isc-dhc-*server* para definir la habilitación y el reinicio del servicio. En este archivo se adjunta las siguientes líneas de código, realizado estos parámetros guardar y cerrar el archivo.

```
package "isc-dhcp-server" do
  action :install
end

service "isc-dhcp-server" do
  supports :status => true, :reload => true, :restart => true
  action [:enable, :start]
end
```

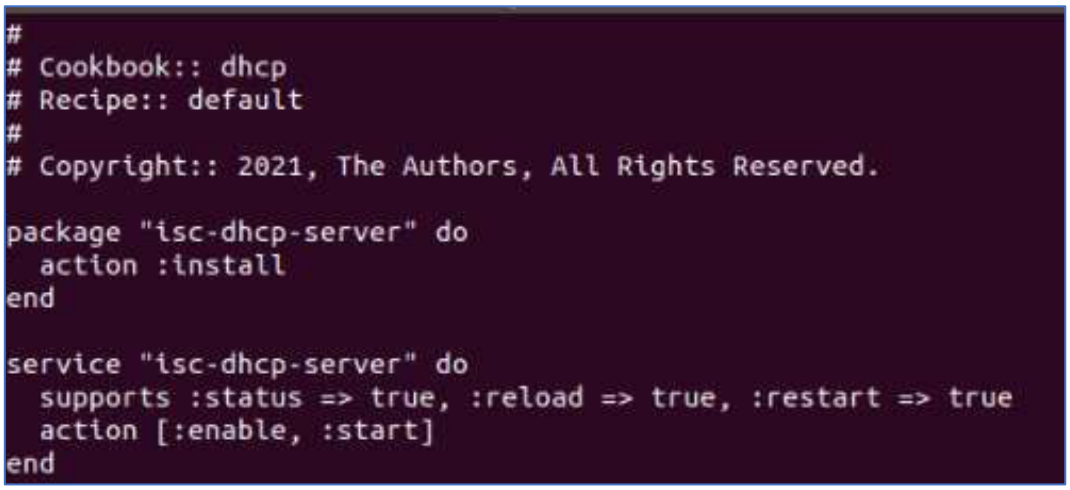


**Figura 3.59** Instalación, habilitación y reinició de servicio dhcp

En la Figura 3.60 se presenta la creación del *template*, *el* cual se ejecutará con la siguiente línea de código para la creación de la plantilla **isc-dhcp-*server*.rb**,

```
:~/chef-repo/cookbooks/dhcp$ chef generate template isc-dhcp-server
```


**Figura 3.60** Creación de *template* . isc-dhcp-*server*

A continuación, ingresar al archivo **isc-dhcp-*server*.erb**, el cual lleva la información de valores pertenecientes a la tarjeta de red donde los nombres de las variables creadas serán definidos en el archivo de receta **default.rb** para el momento de la llamada de ejecución.

En la Figura 3.61 se presenta la modificación de contenido del archivo **isc-dhcp-*server*.erb** en el mismo se definirá la interfaz de red respectiva, la misma que estará disponible para la distribución y asignación de direcciones IP con las que va a trabajar el servidor DHCP, para esto se asignará la interfaz ens39 anteriormente creada.

```
$ sudo vi chef-repo/cookbooks/dhcp/templates/isc-dhcp-server.erb
```

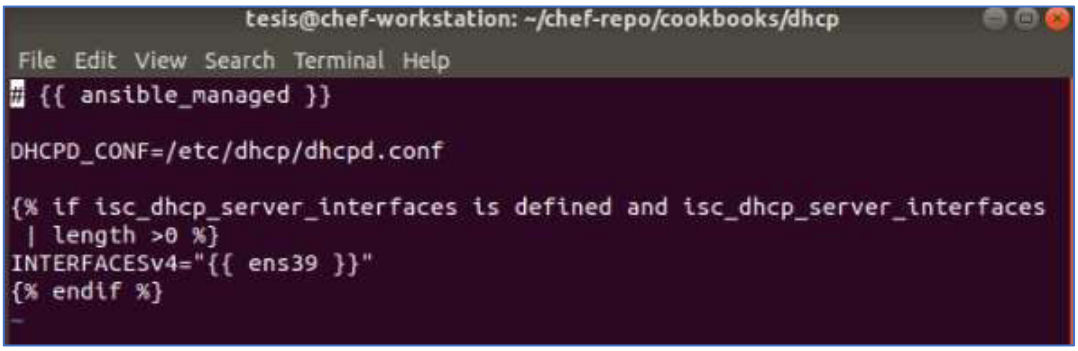


**Figura 3.61** Configuraciones *template* isc-dhcp-*server*.erb

Acto seguido, crear el archivo de plantilla que tendrá el nombre **dhcpd.conf.** Como se presenta en la Figura 3.62 la creación del *template* tiene que estar dentro del directorio *cookbook* dhcp mediante la siguiente línea de código.

```
:~/chef-repo/cookbooks/bind9$ chef generate template dhcpd.conf
```

```
tesis@chef-workstation:~/chef-repo/cookbooks/dhcp$ chef generate template
dhcpd.conf
Recipe: code_generator::template
  * directory[/home/tesis/chef-repo/cookbooks/dhcp/templates] action creat
e (up to date)
  * template[/home/tesis/chef-repo/cookbooks/dhcp/templates/dhcpd.conf.erb
] action create
    - create new file /home/tesis/chef-repo/cookbooks/dhcp/templates/dhcpd
.conf.erb
    - update content in file /home/tesis/chef-repo/cookbooks/dhcp/template
s/dhcpd.conf.erb from none to e3b0c4
    (diff output suppressed by config)
```

**Figura 3.62** Creación de template dhcpd.conf

Creada la plantilla, se ingresará al archivo **dhcpd.conf.erb**, en este archivo se configurará los valores con los que trabajará el servidor DHCP, como se observa en la Figura 3.63 según la configuración requerida se ingresa la dirección que realizará *subneting* con su respectiva máscara, el rango con el que va a proporcionar las direcciones el servidor y la dirección de broadcast.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/dhcp/
template/dhcpd.conf.erb
```

```
# Subnet declarations
#
{% for ip, options in isc_dhcp_server_subnets.items() %}
subnet {{  192.168.247.0}} netmask {{ 255.255.255.0}} {
{% if options.range_begin is defined and options.range_end is defined %}
  range {{192.168.247.5}} {{192.168.247.50}};
{% endif %}
{% if options.routers is defined %}
  option routers {{ 192.168.247.2}};
{% endif %}
{% if options.broadcast_address is defined %}
  option broadcast-address {{ 192.168.247.255 }};
{% endif %}
{% if options.default_lease_time is defined %}
  default-lease-time {{ 600 }};
{% endif %}
{% if options.max_lease_time is defined %}
  max-lease-time {{ 7200 }};
{% endif %}
{% for option in options.other_flags | default([]) %}
  {{ flag }};
{% endfor %}
{% for option in options.other_options | default([]) %}
  option {{ option }};
{% endfor %}
}
```

**Figura 3.63** Configuración y edición de archivo dhcpd.conf

Se prosigue a la modificación de la receta **default.rb**, en la cual se agregará el llamando a las plantillas anteriormente creadas y editadas, con la estructura del lenguaje de

programación Ruby como se indica en la Figura 3.64. Este contenido llevará la ubicación de host, nombre del archivo template que tiene la configuración y los permisos para su ejecución.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/dhcp/
recipes/default.rb
```

```
template "/etc/default/isc-dhcp-server" do
source "isc-dhcp-server.erb"
  owner "root"
  group "root"
  mode 0644
end

template "/etc/dhcp/dhcpd.conf" do
source "dhcpd.conf.erb"
  owner "root"
  group "root"
  mode 0644
end
```

```
#
# Cookbook:: dhcp
# Recipe:: default
#
# Copyright:: 2021, The Authors, All Rights Reserved.

package "isc-dhcp-server" do
  action :install
end

service "isc-dhcp-server" do
  supports :status => true, :reload => true, :restart => true
  action [:enable, :start]
end

template "/etc/default/isc-dhcp-server" do
source "isc-dhcp-server.erb"
  owner "root"
  group "root"
  mode 0644
end

template "/etc/dhcp/dhcpd.conf" do
source "dhcpd.conf.erb"
  owner "root"
  group "root"
  mode 0644
end
"recipes/default.rb" 29 lines, 502 characters
```

**Figura 3.64** Definición de variables de ejecución de template para dhcp

Posteriormente, configurados todos los parámetros descritos en el *cookbook* dhcp, se verifica mediante el comando tree la forma recursiva de los directorios y archivos, como se indica en Figura 3.65, este mostrará el directorio y archivos template creados con anterioridad. Acto seguido, con ayuda de la herramienta *knife,* se subirá el *cookbook* al servidor, mediante la siguiente línea de código; como se muestra en la Figura 3.66 al subir el libro se presentará un mensaje indicando que el *cookbook* fue subido con éxito.

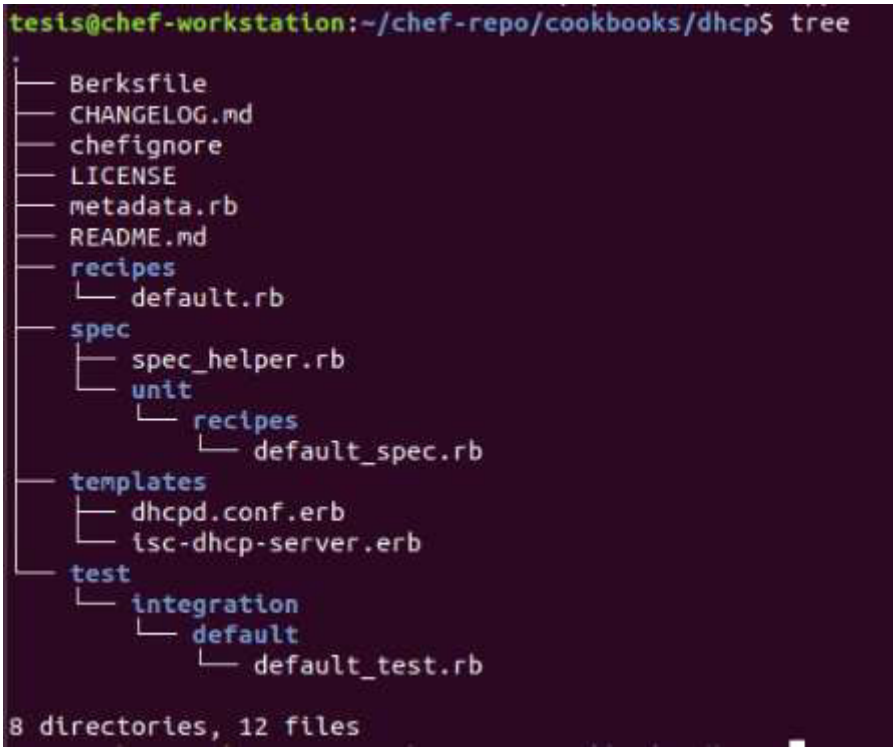


**Figura 3.65** Estructura final de libro de cocina dhcp

tesis@chef-*workstation*:~/chef-repo/*cookbook*s$ knife *cookbook* upload dhcp


**Figura 3.66** *Cookbook* dhcp subido al servidor

Finalmente, se muestra en la Figura 3.67 que se ingresa a la interfaz de Chef Manage y se editará la lista de ejecución en el nodo **chefadmin**, en recetas disponibles se buscará la receta dhcp, encontrada se trasladará a la lista de ejecución actual, después de guarda los cambios estos se podrán visualizar cuando se ingrese en el nodo cliente.

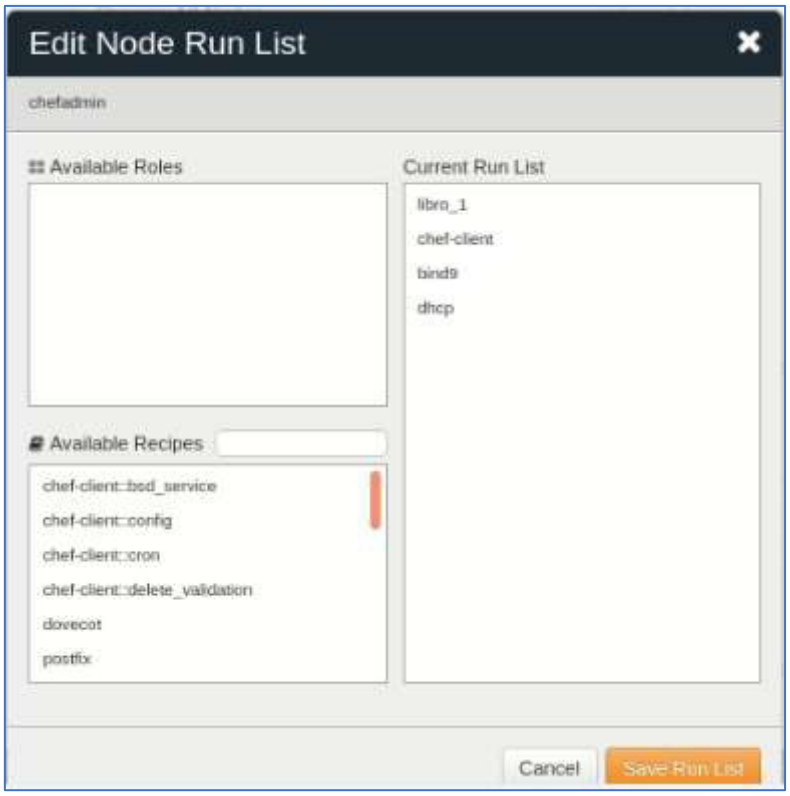


**Figura 3.67** Ejecución de *cookbook* dhcp

### Despliegue del servidor de correo.

Para la instalación del servidor de correo se requerirá el uso de paquetes como Postfix, Dovecot así como también Thunderbird. Estos paquetes contienen servicios de SMTP, IMAP y MUA como gestor de correo [23].

El servidor SMTP es aquel que está encargado del transporte del correo electrónico, el mismo que trabaja bajo el proceso de autenticación en el envió de los paquetes desde el remitente al destino por medio del protocolo de transporte SMTP, este protocolo se lo encontrará en el paquete de Postfix que trabajará como agente de transferencia o MTA [24].

Como agente de reparto o MDA se tiene al paquete Dovecot que trabaja con el protocolo de IMAP, este permite acceder a los mensajes de correo que se encuentran almacenados en el servidor y poder acceder a la bandeja de correo desde otro dispositivo que no sea el principal [25].

Thunderbird trabaja como el agente de usuario gráfico o MUA, el mismo que presentará los mensajes enviados y recibidos, así como también la bandeja de correos, dando un entorno de operación más amigable con el usuario para la interacción y manejo del servicio de correo [22].

En este despliegue se requerirá encontrarse en la estación de trabajo donde se creará libros de cocina para Postfix y Dovecot, estos contendrán las recetas de instalación y configuración para el servidor de correo. Mediante las siguientes líneas de comando se establecerá los nuevos *cookbook*, como se visualiza en la Figura 3.68 y la Figura 3.69 estas tendrán directorios y ficheros por defecto que se crean al momento de la creación de los *cookbooks*. Dirigirse al directorio chef-repo/*cookbooks* donde creará el libro de cocina titulado postfix seguido del libro de cocina dovecot, se verá la estructura creada de cada *cookbook* mediante el comando de árbol el cual se puede ver en la **Figura *3.1***Figura 3.70 y Figura 3.71.

```
tesis@chef-workstation:~$ cd chef-repo/cookbooks
tesis@chef-workstation:~/chef-repo/cookbooks$ chef generate
cookbook postfix
tesis@chef-workstation:~/chef-repo/cookbooks$ chef generate
cookbook dovecot
```

```
tesis@chef-workstation:~/chef-repo/cookbooks$ chef generate cookbook postf
ix
Generating cookbook postfix
- Ensuring correct cookbook file content
- Ensuring delivery configuration
- Ensuring correct delivery build cookbook content

Your cookbook is ready. Type `cd postfix` to enter it.

There are several commands you can run to get started locally developing a
nd testing your cookbook.
Type `delivery local --help` to see a full list.

Why not start by writing a test? Tests for the default recipe are stored a
t:

test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:

recipes/default.rb
```

**Figura 3.68** Creación de libro de cocina postfix

```
tesis@chef-workstation:~/chef-repo/cookbooks$ chef generate cookbook dovec
ot
Generating cookbook dovecot
- Ensuring correct cookbook file content
- Ensuring delivery configuration
- Ensuring correct delivery build cookbook content

Your cookbook is ready. Type `cd dovecot` to enter it.

There are several commands you can run to get started locally developing a
nd testing your cookbook.
Type `delivery local --help` to see a full list.

Why not start by writing a test? Tests for the default recipe are stored a
t:

test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:

recipes/default.rb
```

**Figura 3.69** Creación de libro de cocina dovecot

```
tesis@chef-workstation:~/chef-repo/cookbooks$ cd postfix/
tesis@chef-workstation:~/chef-repo/cookbooks/postfix$ tree
```

```
tesis@chef-workstation:~/chef-repo/cookbooks/postfix$ cd ..
tesis@chef-workstation:~/chef-repo/cookbooks$ cd dovecot/
tesis@chef-workstation:~/chef-repo/cookbooks/dovecot$ tree
```

```
tesis@chef-workstation:~/chef-repo/cookbooks/postfix$ tree
.
├── Berksfile
├── CHANGELOG.md
├── chefignore
├── LICENSE
├── metadata.rb
├── README.md
├── recipes
│   └── default.rb
├── spec
│   ├── spec_helper.rb
│   └── unit
│       └── recipes
│           └── default_spec.rb
└── test
    └── integration
        └── default
            └── default_test.rb

7 directories, 10 files
```

**Figura 3.70** Estructura inicial de libro de cocina postfix

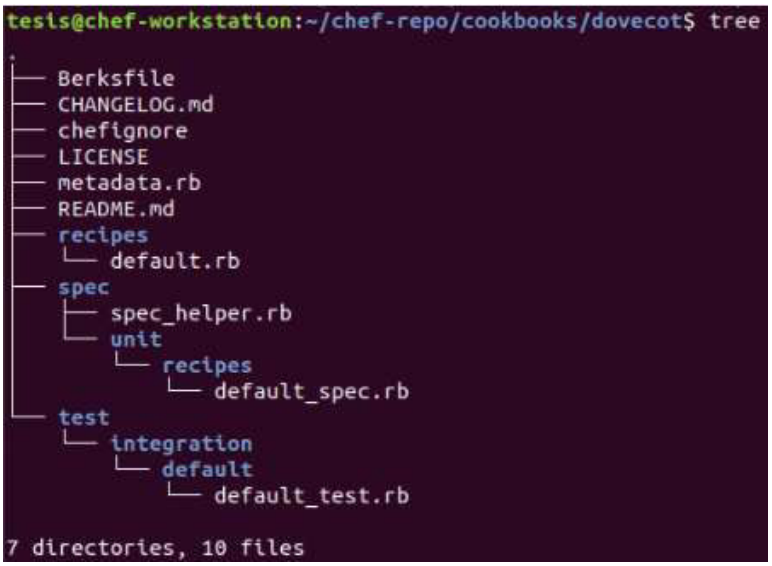


**Figura 3.71** Estructura inicial de libro de cocina dovecot.

Ingresando al directorio de recipes el cual contiene el archivo **defaul.rb** en el *cookbook* postfix, se escribirá la estructura para la instalación del paquete de Postfix adicional a esto se configurará las líneas de código que editarán y llamarán a la plantilla de configuración del **archivo main.cf**.

```
tesis@chef-workstation:~/chef-repo/cookbooks/postfix$ sudo vi
recipes/default.rb
```

En la Figura 3.72 se observará que se define el recurso del paquete para instalar, seguido se llama al paquete postfix para definir la habilitación y el reinicio del servicio. En este archivo se adjunta adicionalmente las siguientes líneas de código para la instalación de utilidades que permitirán enviar correos y entran en el buzón para la lectura de estos, terminada la edición de los parámetros guardar y cerrar el archivo.

```
package "postfix" do
  action :install
end

service "postfix" do
  supports :status => true, :reload => true, :restart => true
  action [:enable, :start]
end

package "mailutils" do
  action :install
end
```

```
package "postfix" do
  action :install
end

service "postfix" do
  supports :status => true, :reload => true, :restart => true
  action [:enable, :start]
end

package "mailutils" do
  action :install
end
```

**Figura 3.72** Instalación, habilitación y reinició de servicio Postfix

Como se presenta en la Figura 3.73 se realiza la creación del template, el cual se ejecutará con la siguiente línea de código para la creación de la plantilla **main.cf**.

```
:~/chef-repo/cookbooks/postfix$ chef generate template main.cf
```

```
tesis@chef-workstation:~/chef-repo/cookbooks/postfix$ chef generate templa
te main.cf
Recipe: code_generator::template
  * directory[/home/tesis/chef-repo/cookbooks/postfix/templates] action cr
eate
    - create new directory /home/tesis/chef-repo/cookbooks/postfix/templat
es
  * template[/home/tesis/chef-repo/cookbooks/postfix/templates/main.cf.erb
] action create
    - create new file /home/tesis/chef-repo/cookbooks/postfix/templates/ma
in.cf.erb
    - update content in file /home/tesis/chef-repo/cookbooks/postfix/templ
ates/main.cf.erb from none to e3b0c4
    (diff output suppressed by config)
```

**Figura 3.73** Creación de template main.cf

Tras haber creado el *template*, ingresar al archivo **main.cf.erb**, el cual lleva la información correspondiente que se editará mediante líneas de código en el archivo **default.rb**.

En este archivo **main.cf.erb** se tendrá parámetros de interés como el *myhostname* que es el nombre largo del equipo (FQDN) que se configuró en el DNS. Con este parámetro se podrá identificar al servidor de correo, *inet_interfaces, inet_protocols* y para finalizar el *home_mailbox*, cada uno de ellos se encuentra especificado en la plantilla **main.cf.erb** que está escrita en lenguaje de programación Ruby como se muestra en la Figura 3.74.

```
$ sudo vi chef-repo/cookbooks/dhcp/templates/main.cf.erb
```

```
biff = no
append_dot_mydomain = no
<% if node['postfix']['smtpd_use_tls'] == "yes" -%>
smtpd_tls_security_level = may
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
<% end -%>
<% if node['postfix']['smtp_use_tls'] == "yes" -%>
smtp_tls_security_level = may
<% end -%>
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtp_sasl_auth_enable = <%= node['postfix']['smtp_sasl_auth_enable'] %>
<% if node['postfix']['smtp_sasl_auth_enable'] == "yes" -%>
smtp_sasl_password_maps = <%= node['postfix']['smtp_sasl_password_maps'] %
>
smtp_sasl_security_options = <%= node['postfix']['smtp_sasl_security_optio
ns'] %>
smtp_tls_CAfile = <%= node['postfix']['smtp_tls_cafile'] %>
<% end -%>
myhostname = <%= node['postfix']['myhostname'] %>
<% if node['postfix']['smtpd_sasl_auth_enable'] == "yes" -%>
smtpd_sasl_auth_enable = yes
```

**Figura 3.74** Archivo template main.cf

Se prosigue a la modificación de la receta **default.rb**. Dentro de este fichero se encuentran los comandos necesarios mediante los cuales se brindarán los permisos para que se ejecuten los servicios sobre todas las interfaces y receptando todos los protocolos previamente descritos. Adicionalmente, se añade el directorio llamado /**Maildir** dentro del cual se almacenarán todos los mensajes, este procedimiento se aprecia en la Figura 3.75.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/postfix/
recipes/default.rb
```

```
package "postfix" do
  action :install
end

service "postfix" do
  supports :status => true, :reload => true, :restart => true
  action [:enable, :start]
end

package "mailutils" do
  action :install
end


template "/etc/postfix/main.cf" do
source "main.cf.erb"
default['postfix']['myhostname'] = node['fqdn']
default['postfix']['inet_interfaces'] = all
default['postfix']['inet_protocols'] = all
default['postfix']['home_mailbox'] = 'Maildir/'
end
```

**Figura 3.75** parámetros de fichero main.cf

Acto seguido, crear las plantillas de configuración para el servicio Dovecot que tendrá el nombre **10-auth.conf** y **10-mail.conf.** Como se presenta en la Figura 3.76 la creación de los t*emplates* tienen que estar dentro del directorio *cookbook* dovecot mediante la siguiente línea de código.

```
:~/chef-repo/cookbooks/dovecot$ chef generate template 10-auth.conf
:~/chef-repo/cookbooks/dovecot$ chef generate template 10-mail.conf
```

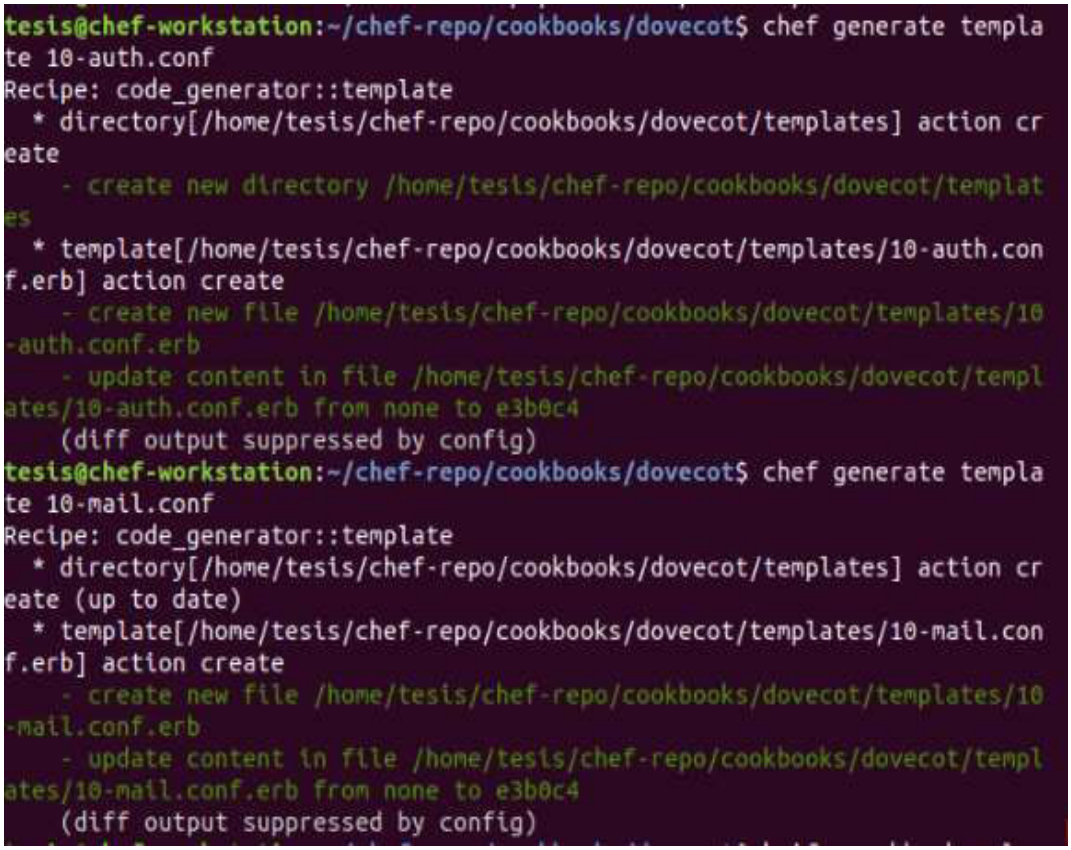


**Figura 3.76** Creación de templates en Dovecot

Creada la plantilla, se ingresará al archivo **10-auth.conf.erb**, en este se configurará el parámetro de permisos para contraseñas en texto plano, como se observa en la Figura 3.77 este archivo posteriormente será editado desde el **default.rb** para su llamado y ejecución.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/dovecot/
template/10-auth.conf.erb
```


**Figura 3.77** Template 10-auth.conf.erb para configuración Dovecot

A continuación, se ingresará al archivo **10-mail.conf.erb**, como se presenta en la Figura 3.78 donde se especificará cual buzón de almacenamiento utilizará Dovecot. Esta configuración se realizó en Postfix, también debido a que los dos paquetes deben manejar la misma dirección de buzón de almacenamiento, como se observa en este archivo posteriormente será editada desde el **default.rb** para su llamado y ejecución.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/dovecot/
template/10-auth.conf.erb
```

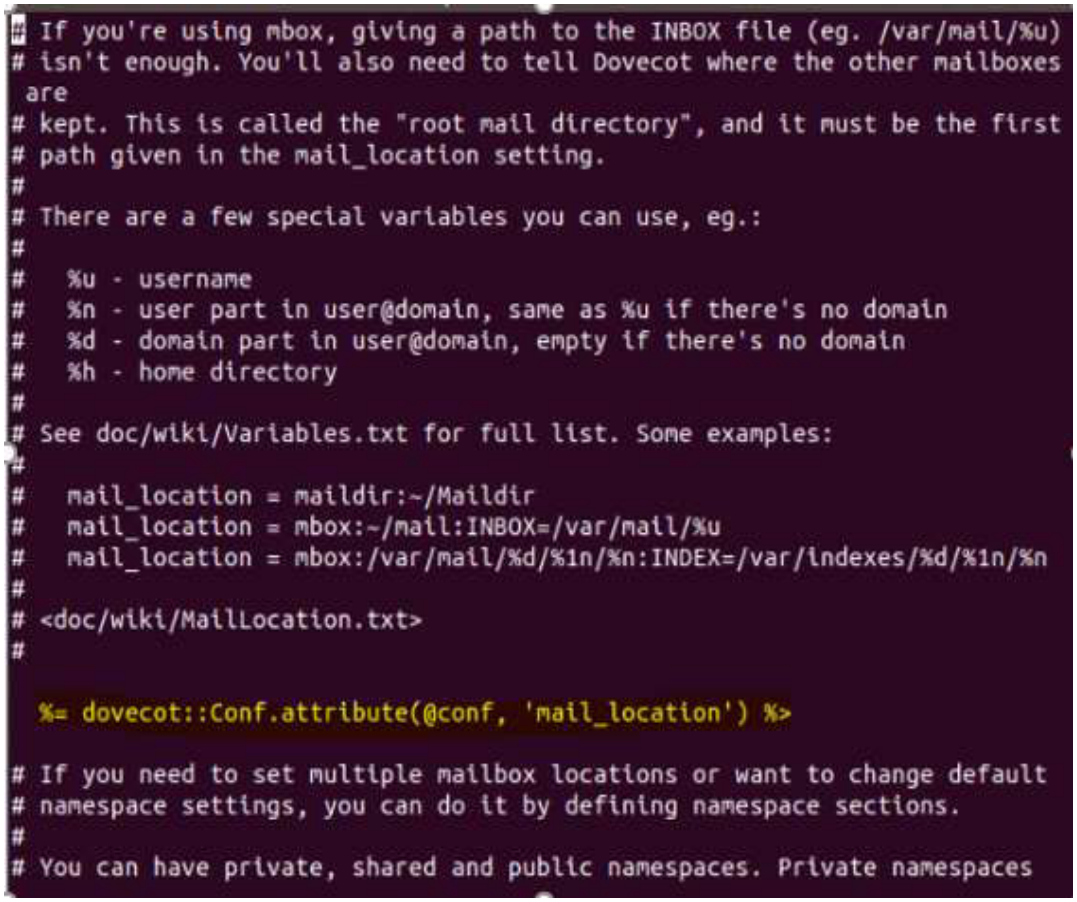


**Figura 3.78** Template 10-mail.conf.erb para configuración Dovecot

Se prosigue a la modificación de la receta **default.rb**, en la cual se ejecutarán las siguientes líneas de código para la instalación del paquete Dovecot, así mismo se agregará el llamando a las plantillas anteriormente creadas y editadas, como se indica en la Figura 3.79 este contenido llevará la ubicación de host, nombre del archivo template que tiene la configuración y los permisos para su ejecución.

```
tesis@chef-workstation:~$ sudo vi chef-repo/cookbooks/dovecot/
recipes/default.rb
```

```
package "dovecot-core" do
  action :install
end

# 10-auth.conf
template "/etc/dovecot/conf.d/10-auth.conf" do
source "10-auth.conf.erb"
node.default['dovecot']['conf']['disable_plaintext_auth'] = no
end


# 10-mail.conf
template "/etc/dovecot/conf.d/10-mail.conf" do
source "10-mail.conf.erb"
node.default['dovecot']['conf']['mail_location'] = 'maildir:~/Maildir'
end
```


**Figura 3.79** Definición de variables de instalación y ejecución de *templates* para dovecot

Posteriormente, configurado todos los parámetros descritos en los *cookbooks* postfix y dovecot, se verifica mediante el comando tree la forma recursiva de los directorios y archivos para los dos libros de cocina, como se indica en Figura 3.80 y la Figura 3.81.

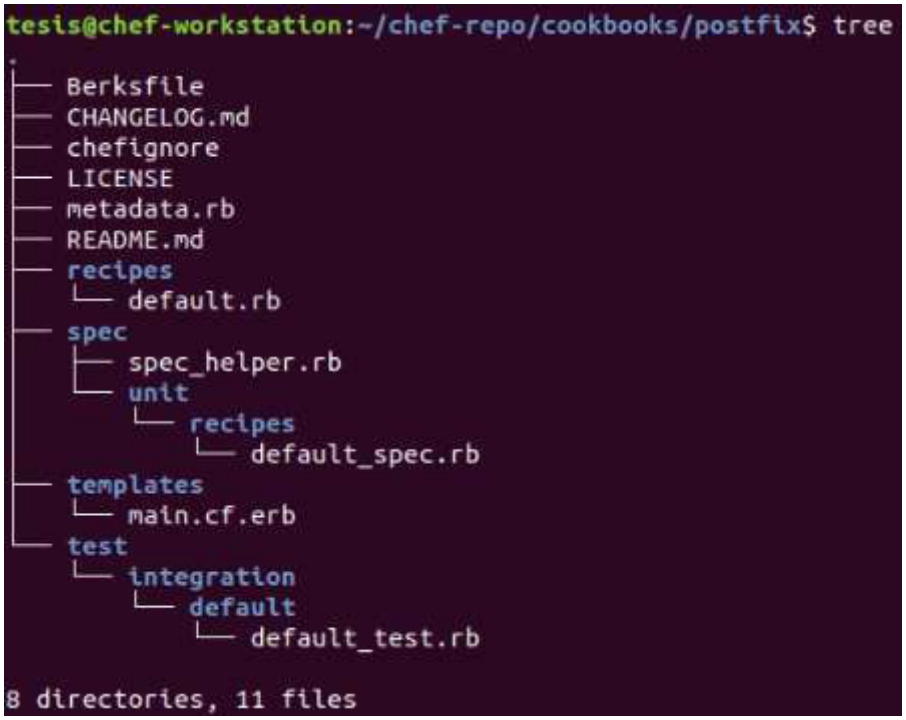


**Figura 3.80** Estructura final de libro de cocina postfix

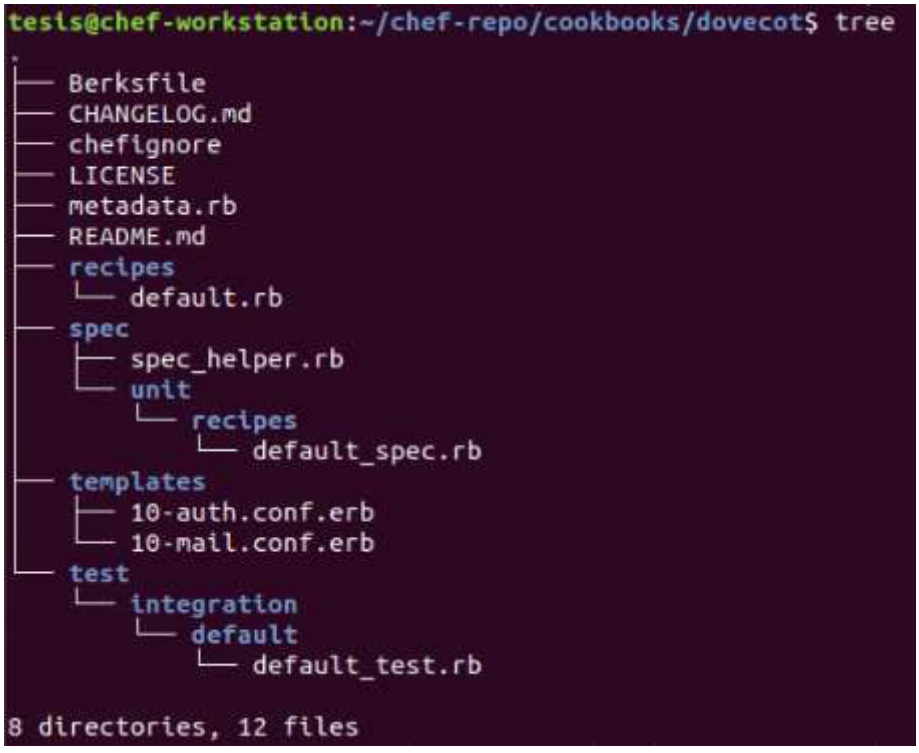


**Figura 3.81** Estructura final de libro de cocina dovecot

Acto seguido, con ayuda de la herramienta *knife* se subirá los *cookbook* al servidor, mediante las siguientes líneas de código; como se muestra en la Figura 3.82 y al subir el libro se presentará un mensaje indicando que el *cookbook* fue subido correctamente.

tesis@chef-*workstation*:~/chef-repo/*cookbook*s$ knife *cookbook* upload postfix

tesis@chef-*workstation*:~/chef-repo/*cookbook*s$ knife *cookbook* upload dovecot

```
tesis@chef-workstation:~/chef-repo/cookbooks$  knife cookbook upload postf
ix

Uploading postfix         [0.1.0]
Uploaded 1 cookbook.
tesis@chef-workstation:~/chef-repo/cookbooks$ knife cookbook upload doveco
t
Uploading dovecot         [0.1.0]
Uploaded 1 cookbook.
```

**Figura 3.82** *Cookbook* postfix y dovecot subidos al servidor

Finalmente, como se muestra en la Figura 3.83 que se ingresa a la interfaz de Chef Manage y se editará la lista de ejecución en el nodo **chefadmin**, en la sesión de recetas disponibles se buscará las recetas de postfix y dovecot, encontradas las mismas se trasladarán a la lista de ejecución actual. Después de guardar los cambios estos se podrán visualizar cuando se ingrese en el nodo cliente.

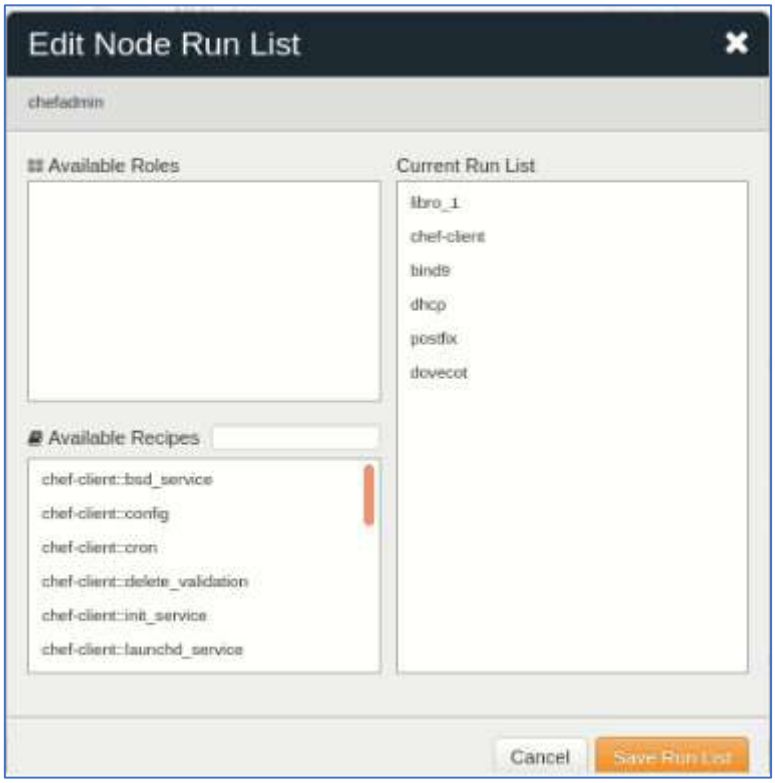


**Figura 3.83** Ejecución de *cookbooks* postfix y dovecot

## 3.4 Verificar los resultados obtenidos con Chef

Para comenzar con el análisis de resultados se deben tener en cuenta diferentes consideraciones, una de las más importantes es verificar que la versión de Chef que se encuentra instalado en el nodo cliente, sea compatible con la versión de Chef que fue instalado en la *Workstation*, así también tener instalado la versión más actual del lenguaje de programación Ruby.

### Verificación del funcionamiento del servidor DNS

En este apartado se ingresará a la máquina del nodo cliente, en la misma se iniciará la ejecución de los libros de cocina con sus respectivas recetas, esto se realiza con el comando presentado posteriormente, en la Figura 3.84 se muestra la ejecución y configuración mediante *cookbooks*, la Figura 3.85 mostrará la instalación de bind9 y configuración de los respectivos directorios y archivos anteriormente descritos.

```
tesis@chef-client:~$ sudo chef-client
```

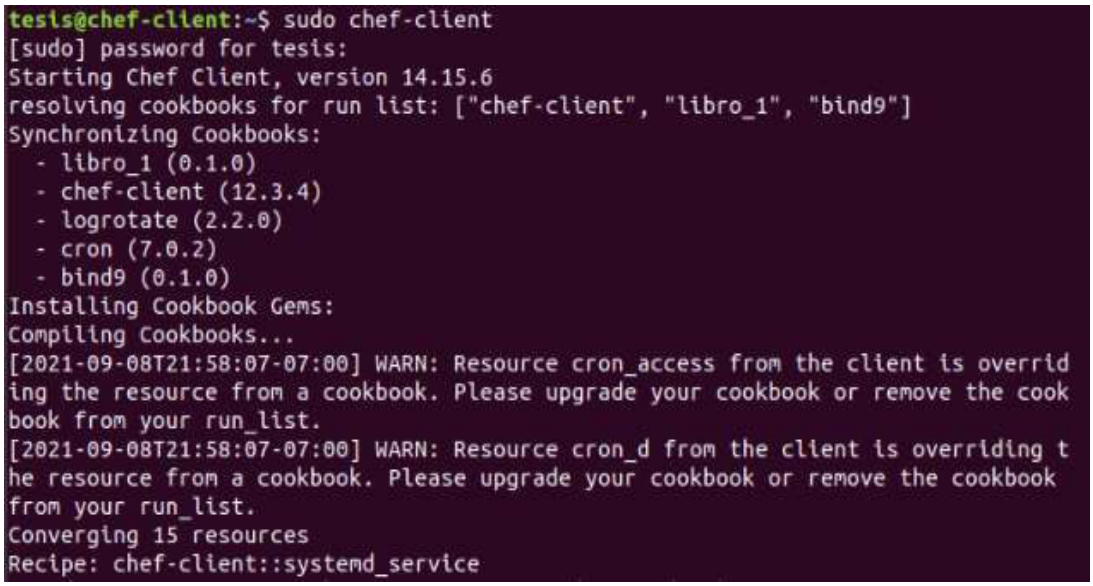


**Figura 3.84** Ejecución de recetas agregadas en nodo cliente

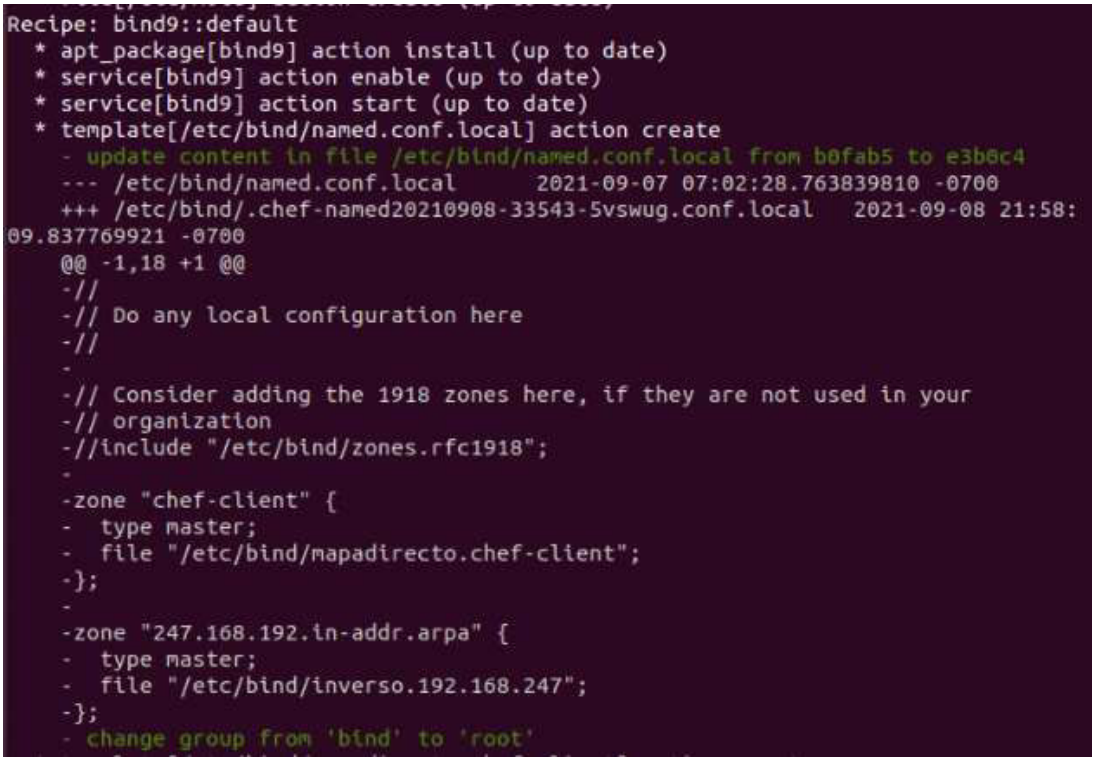


**Figura 3.85** Ejecución bind9 en nodo cliente

Finalizado el proceso de instalación y configuración, se ingresará las siguientes líneas de código donde se reiniciará el servicio de DNS, seguido de la visualización del estado del servidor bind9 que mostrará que todo está cargado y activo esto se puede ver en la Figura 3.86.

```
tesis@chef-client:~$ sudo /etc/init.d/bind9 restart

tesis@chef-client:~$ sudo service bind9 status
```

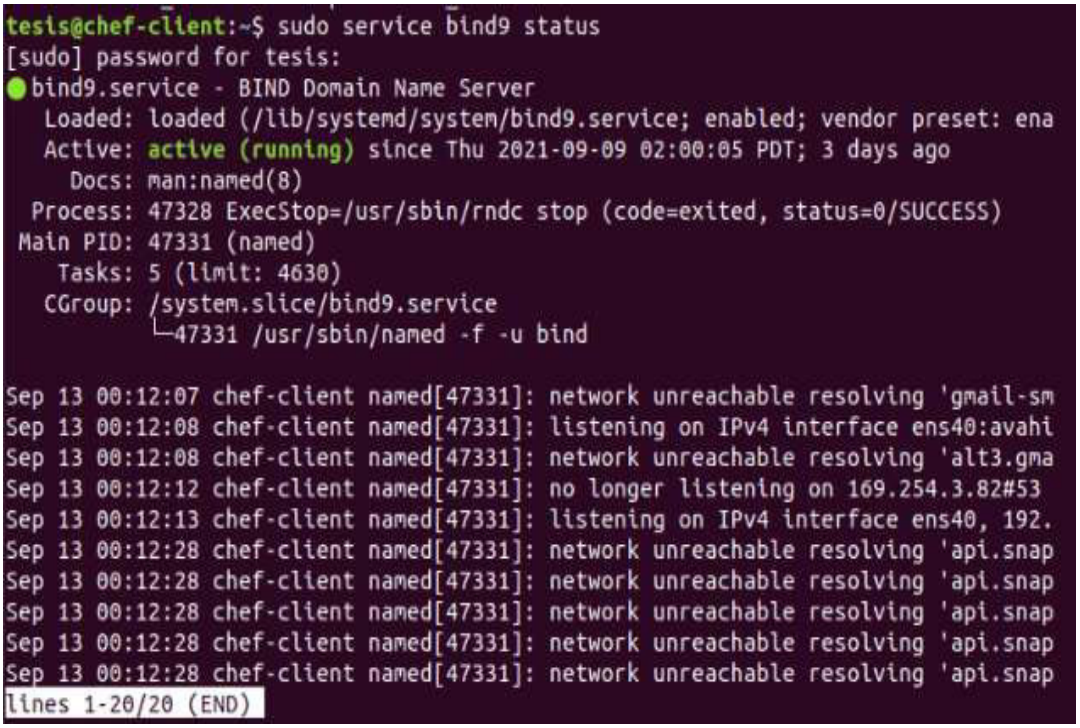


**Figura 3.86** Estado activo de biend9

Seguidamente se realizará una serie de consultas al servidor DNS, para comprobar que el mismo funciona y resuelve las tradiciones entre nombres de dominio y direcciones IP.

Se comprobará mediante un ping con la dirección IP del adaptador de red para examinar si se encuentra activo y funcionando como se presenta en la Figura 3.87, inmediatamente se volverá a realizar un ping con el nombre *server* este automáticamente le añade al nombre relativo configurado en la zona y mostrará cual es la dirección asignada al mismo.

```
tesis@chef-client:~$ ping -c 4 192.168.247.211

tesis@chef-client:~$ ping -c 4 server
```

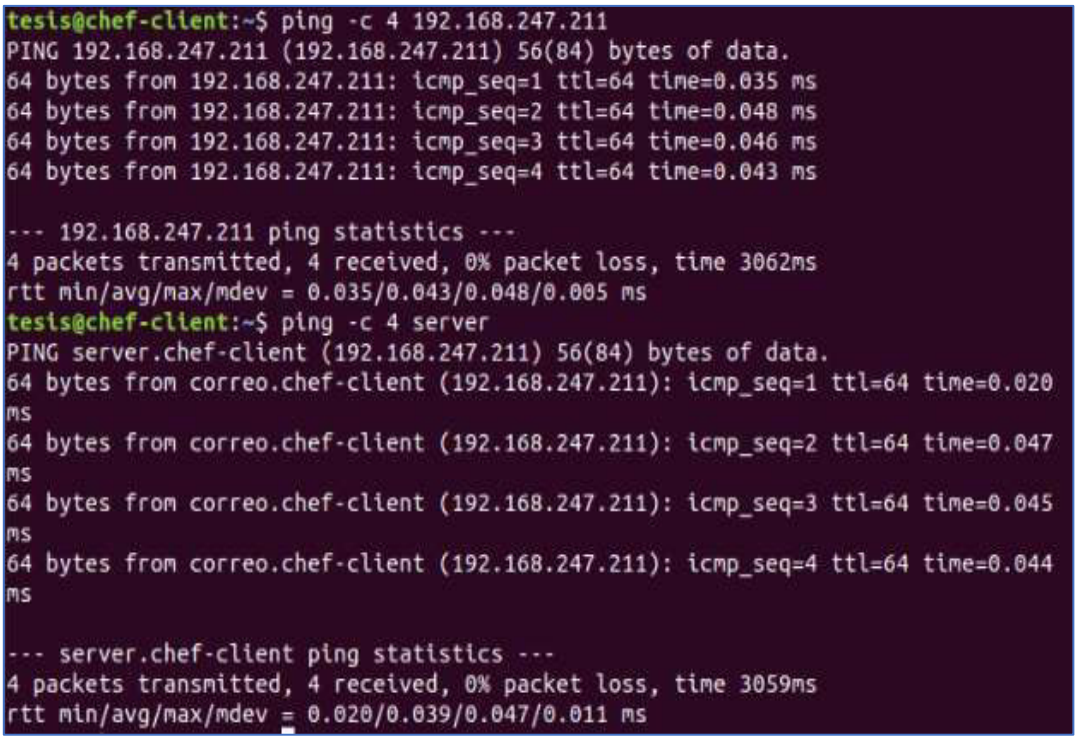


**Figura 3.87** ping de forma directa y recursiva del servicio DNS

Adicionalmente se podrá comprobar con el comando nslookup, que es una herramienta disponible para la verificación de la resolución correcta de direcciones IP a nombres de dominio y viceversa. Con los comandos a continuación se presentará la información detallada de qué servidor está respondiendo a la petición como los nombres de dominio que estén involucrados a la dirección IP, estas consultas se las puede realizar de forma directa o inversa como se visualiza en la Figura 3.88.

```
tesis@chef-client:~$ nslookup server

tesis@chef-client:~$ nslookup 192.168.247.211
```

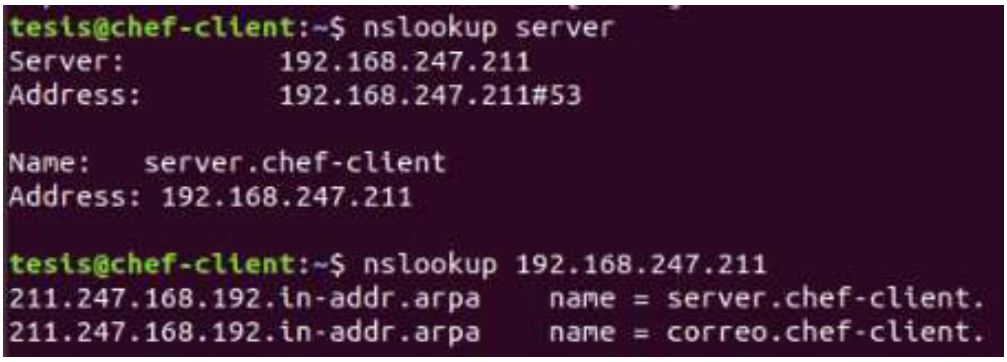


**Figura 3.88** Comprobación del servicio DNS mediante nslookup

### Verificación del funcionamiento del servidor DHCP

Para la sección de la comprobación del servidor DHCP, se ingresará a la máquina del nodo cliente, en la misma se iniciará la ejecución de los libros de cocina como se muestra en la Figura 3.89. Con sus respectivas recetas mediante la línea de código descrita a continuación, se muestra la ejecución y configuración mediante *cookbooks*, en Figura 3.90 mostrará la instalación del paquete isc-dhcp-*server* y la configuración de los respectivos directorios y archivos anteriormente descritos.

```
tesis@chef-client:~$ sudo chef-client
```

```
tesis@chef-client:~$ sudo chef-client
[sudo] password for tesis:
Starting Chef Client, version 14.15.6
resolving cookbooks for run list: ["libro_1", "chef-client", "bind9", "dhcp"]
Synchronizing Cookbooks:
  - libro_1 (0.1.0)
  - cron (7.0.2)
  - chef-client (12.3.4)
  - bind9 (0.1.0)
  - logrotate (2.2.0)
  - dhcp (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
[2021-09-08T22:56:30-07:00] WARN: Resource cron_access from the client is overrid
ing the resource from a cookbook. Please upgrade your cookbook or remove the cook
book from your run_list.
[2021-09-08T22:56:30-07:00] WARN: Resource cron_d from the client is overriding t
he resource from a cookbook. Please upgrade your cookbook or remove the cookbook
from your run_list.
Converging 19 resources
```

**Figura 3.89** Ejecución de nuevas recetas en nodo cliente

```
Recipe: dhcp::default
  * apt_package[isc-dhcp-server] action install (up to date)
  * service[isc-dhcp-server] action enable (up to date)
  * service[isc-dhcp-server] action start (up to date)
  * template[/etc/default/isc-dhcp-server] action create
    - update content in file /etc/default/isc-dhcp-server from 2b381f to e3b0c4
    --- /etc/default/isc-dhcp-server    2021-09-08 12:50:24.089461785 -0700
    +++ /etc/default/.chef-isc-dhcp-server20210908-35074-qcfk1a 2021-09-08 22:56:
32.525046988 -0700
    @@ -1,19 +1 @@
    -# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)
    -
    -# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
    -#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
    -#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf
    -
    -# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
    -#DHCPDv4_PID=/var/run/dhcpd.pid
    -#DHCPDv6_PID=/var/run/dhcpd6.pid
    -
    -# Additional options to start dhcpd with.
    -#  Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
    -#OPTIONS=""
    -
    -# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
    -#  Separate multiple interfaces with spaces, e.g. "eth0 eth1".
    -INTERFACESv4="ens39"
    -INTERFACESv6=""
```

**Figura 3.90** Ejecución dhcp en nodo cliente.

Terminado el proceso de instalación y configuración para el servidor DHCP. Se configurará el cliente DHCP el mismo que se maneja en la interfaz creada anteriormente, este con la identificación de ens40, para la cual se ingresará al archivo de configuración de interfaces ingresando el comando mostrado a continuación, este abrirá el archivo. Se adjuntará las líneas que especifiquen que el adaptador de red ens40 trabajará de forma dinámica con DHCP esto se muestra en la Figura 3.91.

```
tesis@chef-client:~$ sudo nano /etc/network/interfaces
```

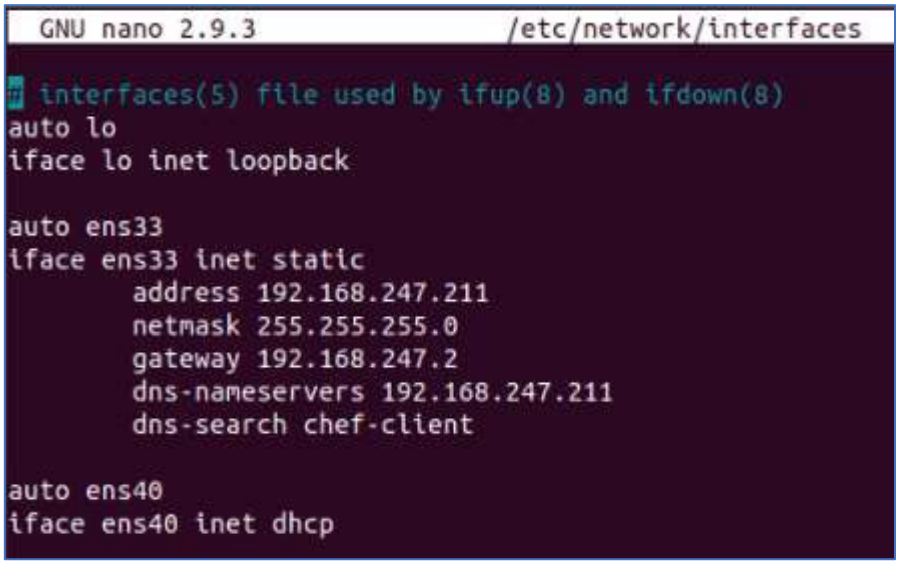


**Figura 3.91** Configuración de interfaz de red para direccionamiento dinámico

Sucesivamente, se ingresan las siguientes líneas de código donde se reiniciará el servicio de DHCP, seguido de la visualización del estado del servidor isc-dhcp-*server* que mostrará que todo está cargado y activo esto se puede ver en la Figura 3.92.

```
tesis@chef-client:~$ sudo systemctl start isc-dhcp-server

tesis@chef-client:~$ sudo systemctl enable isc-dhcp-server

tesis@chef-client:~$ sudo systemctl status isc-dhcp-server
```

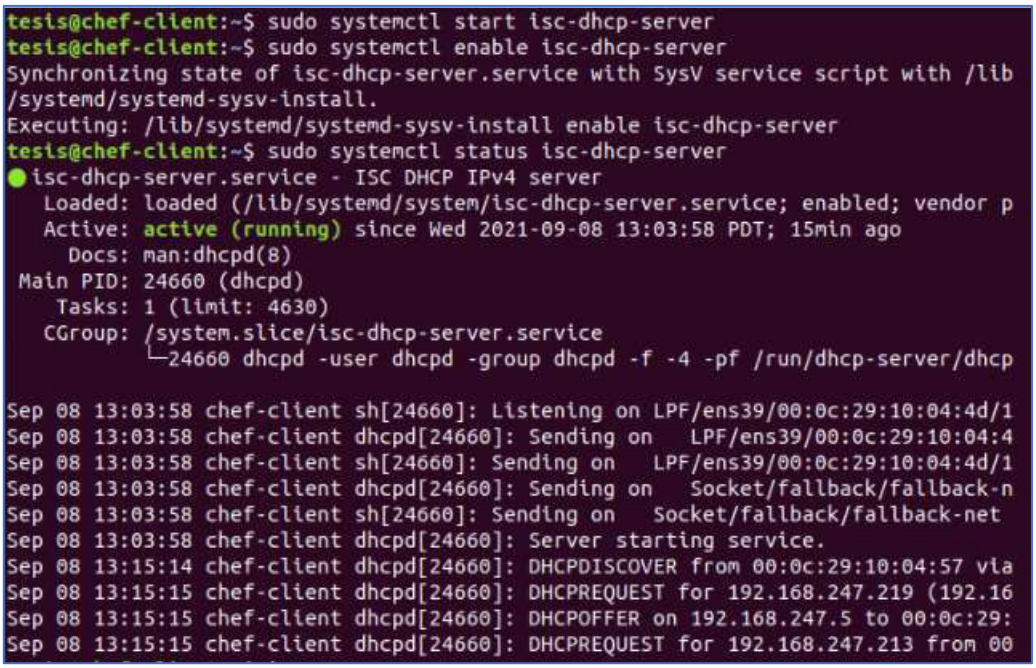


**Figura 3.92** Estado de isc-dhcp-*server* activo

Se reiniciará el adaptador de red para que muestre la negociación de DHCP, ingresando las siguientes líneas de código que desactivarán al adaptador de red ens40 y lo encenderá en acto seguido como se muestra en la Figura 3.93. Se presentará un informe donde se muestra un paquete DHCP en el que comunica que se tomará una nueva dirección y hará uso de la dirección IP proporcionada por el servidor, esta dirección es una de las que se encuentra configurado el rango anteriormente descrito.

```
tesis@chef-client:~$ sudo ifdown ens40

tesis@chef-client:~$ sudo ifup ens40
```

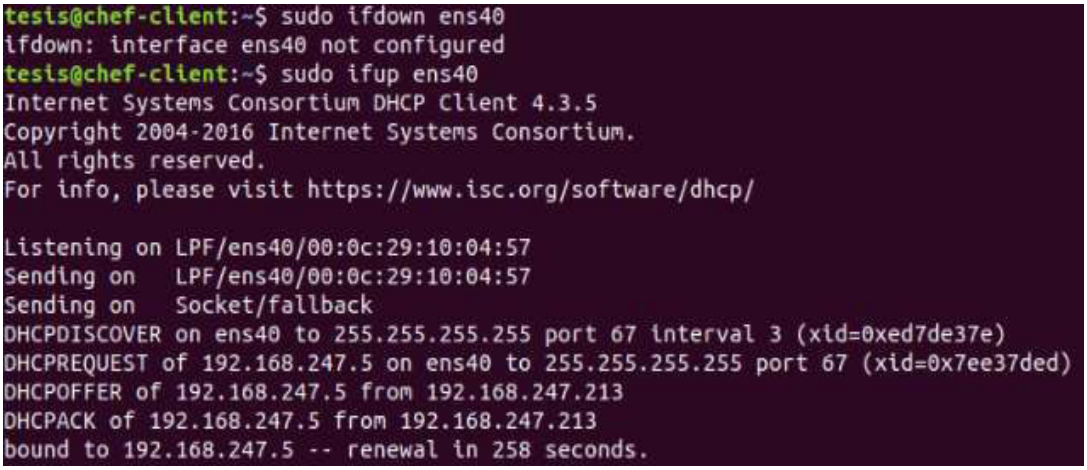


**Figura 3.93** Habilitación de interfaz para DHCP

Se verá la configuración de las interfaces de red que proporcionará el comando descrito a continuación, este desplegará información completa del enrutamiento en la red de los diferentes adaptadores que se encuentran habilitados en la máquina, la Figura 3.94 muestra la dirección dinámica que fue proporcionada al cliente DHCP por el servidor, la misma que trabaja bajo el rango establecido.

```
tesis@chef-client:~$ ip a
```

```
5: ens40: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
    link/ether 00:0c:29:10:04:57 brd ff:ff:ff:ff:ff:ff
    inet 192.168.247.219/24 brd 192.168.247.255 scope global dynamic noprefixrout
e ens40
       valid_lft 1478sec preferred_lft 1478sec
    inet 192.168.247.5/24 brd 192.168.247.255 scope global secondary ens40
       valid_lft forever preferred_lft forever
    inet6 fe80::fd09:f6fe:bb94:3c90/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

**Figura 3.94** Comprobación de dirección dinámica en adaptador cliente

Adicionalmente se podrá comprobar con el comando a continuación las direcciones DHCP asignadas por el servidor, este comando lleva un almacenamiento de datos de todas las direcciones asignadas a las diferentes máquinas que quieran trabajar dinámicamente, ver la Figura 3.95. Se puede observar la primera dirección asignada a la máquina chef-*client*, el tiempo válido de la dirección, e incluso la dirección MAC de la máquina a la que fue proporcionada dicha dirección IP.

```
tesis@chef-client:~$ dhcp-lease-list
```

```
tesis@chef-client:~$ dhcp-lease-list
To get manufacturer names please download http://standards.ieee.org/regauth/oui/o
ui.txt to /usr/local/etc/oui.txt
Reading leases from /var/lib/dhcp/dhcpd.leases
MAC                IP              hostname       valid until         manufacture
r
--------------------------------------------------------------------------------
--------------
00:0c:29:10:04:57  192.168.247.5   chef-client    2021-09-08 20:30:28 -NA-
```

**Figura 3.95** Listado de direcciones asignadas por servidor DHCP

### Verificación del funcionamiento del servidor de correo

Una vez localizados en chef-*client* , se dará inicio a la ejecución de los libros de cocina con las correspondientes recetas , mediante el siguiente comando.

```
tesis@chef-client:~$ sudo chef-client
```

Se puede visualizar en la Figura 3.96 y la Figura 3.97 todos los libros ejecutados anteriormente y las respectivas recetas de postfix y dovecot, donde presentará el resumen de ejecución de cada uno de los procesos que se desea implementar en el nodo cliente.

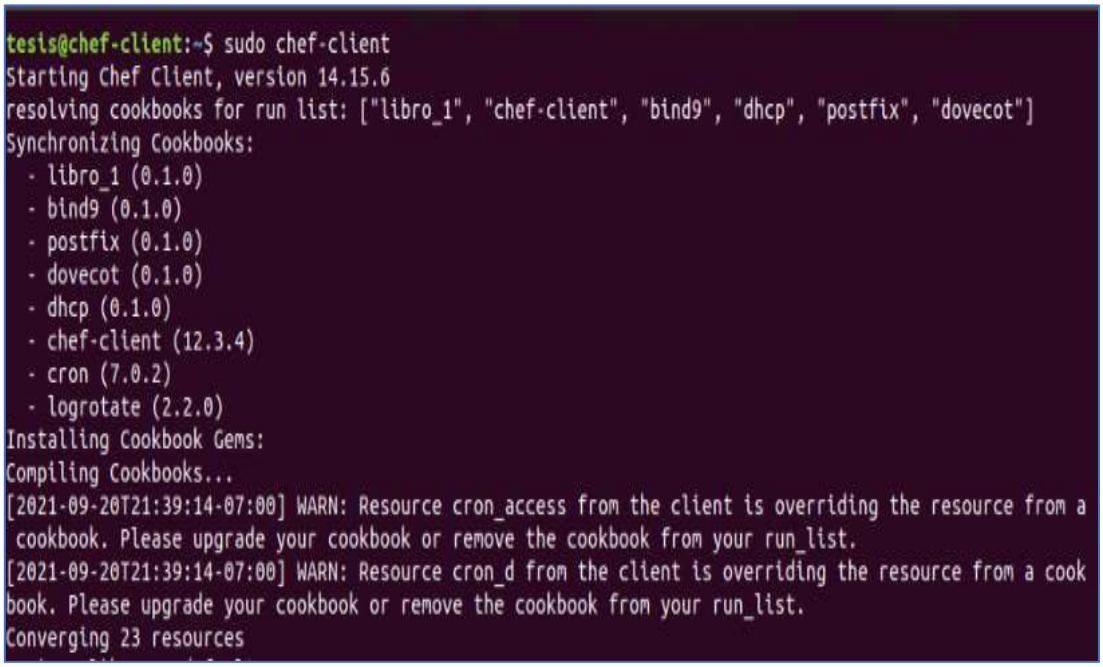


**Figura 3.96** Resolución de libros de cocina


**Figura 3.97** Lista de Ejecución de libros de cocina

Para la comprobación del servidor del correo se necesita crear cuentas de usuarios con el objetivo de poder enviar y recibir correos, estas cuentas se crearán mediante el siguiente comando propuesto a continuación. En la Figura 3.98 no presenta las características de edición para el usuario como nombre, números de teléfono y entre otras opciones, estas se configurarán dependiendo de las características que se desee, todas estas configuraciones se las realizará en el nodo cliente.

```
tesis@chef-client:~$ sudo adduser jenny
```

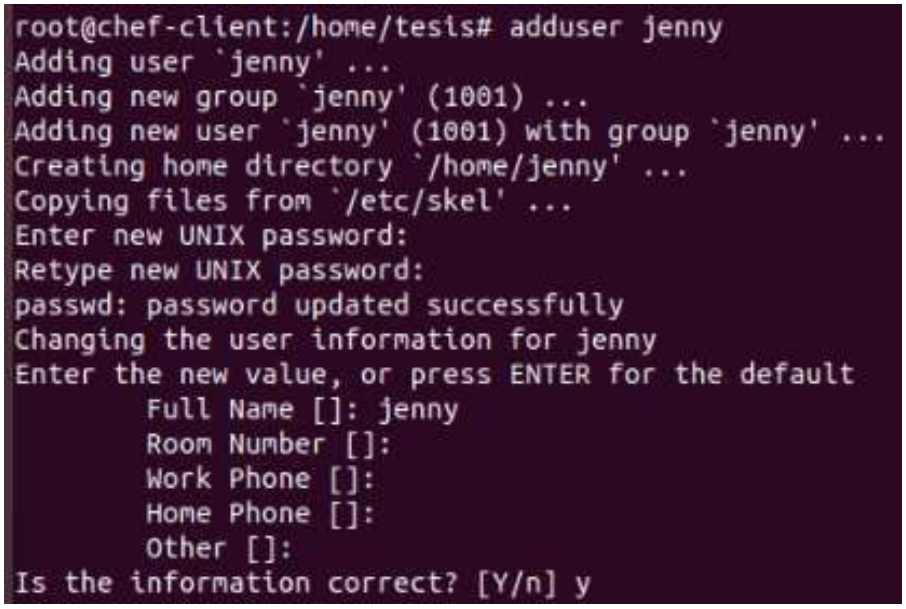


**Figura 3.98** Creación de cuentas de usuarios

Lista la primera cuenta de usuario se repetirá los comandos descritos anteriormente, para la creación de un segundo usuario, con el comando a continuación se verificará todos los usuarios existentes con los que se podrá trabajar esto se podrá visualizar en la Figura 3.99.

```
tesis@chef-client:~$ sudo ls -l /home/
```

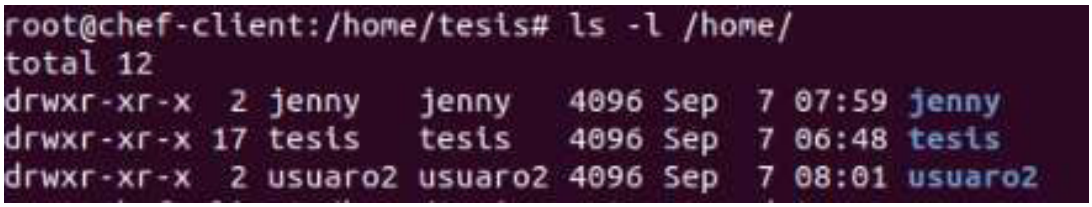


**Figura 3.99** Lista de usuarios existentes

Posteriormente se hará uso del MUA gráfico o Thunderbird, este ya se encuentra instalado y es un paquete por defecto en la versión de Ubuntu 18.04, en la cual permitirá realizar envío y recepción de correos mediante una interfaz gráfica, ingresar la siguiente línea de comando en consola, la misma que ejecutará la aplicación y la abrirá.

```
tesis@chef-client:~$ thunderbird
```

Sucesivamente, para realizar la comprobación se configura dos cuentas de correo, estas cuentas son pertenecientes a los usuarios ya creados anteriormente, esto se realizará manualmente en la pestaña que se abre inmediatamente después de la ejecución de

Thunderbird en consola, se ingresará los parámetros como se muestra en la Figura 3.100. Posteriormente se dará clic en configuración manual, la cual desplegará opciones de configuración para los campos entrantes y salientes con los que se va a manejar la plataforma de correo.

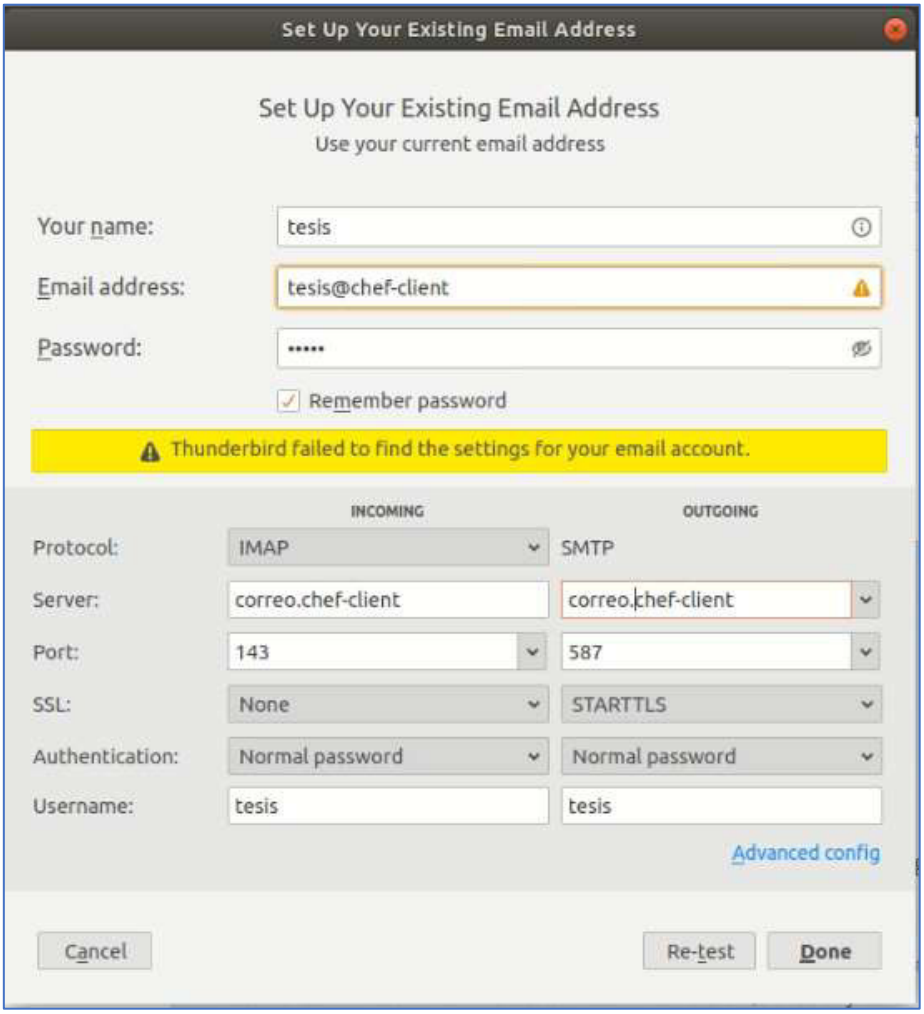


**Figura 3.100** Configuración de dirección de correo electrónico existente

Tras haber configurado los campos se dará clic en Done, en esta instancia se desplegará una nueva ventana como se muestra en la Figura 3.101 en la que se advierte que la configuración de entrada no usa cifrado, seleccionar la opción en la que se entiende los riesgos.

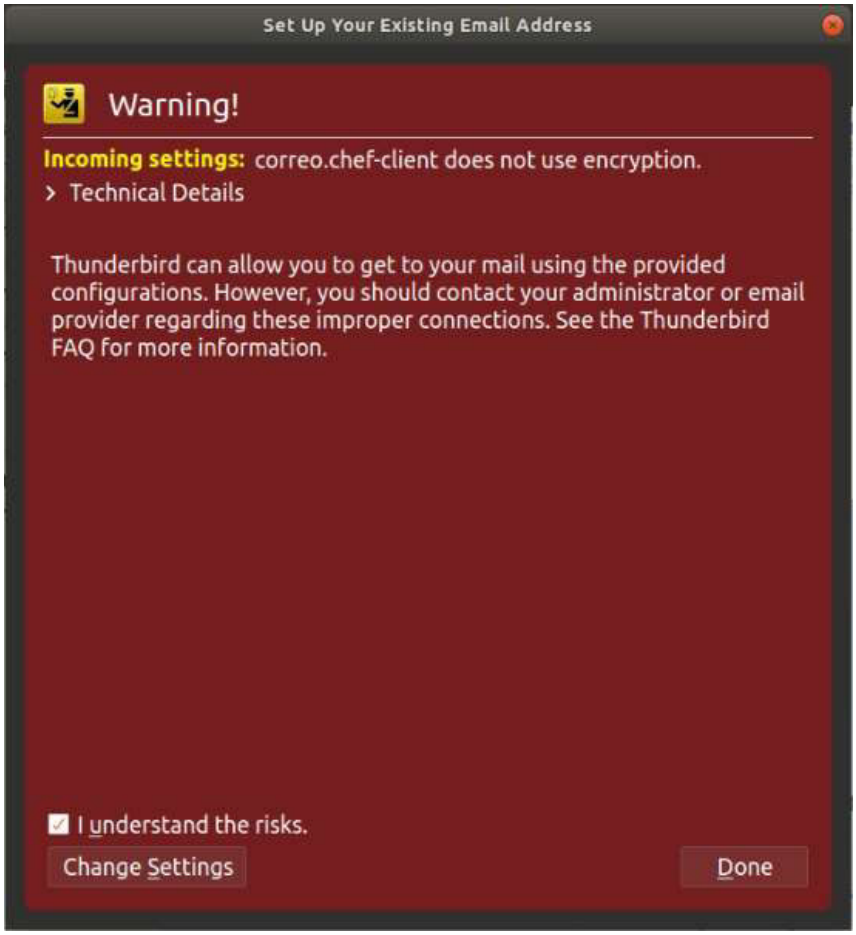


**Figura 3.101** Ventana de advertencia de configuración de entrada

Finalmente, se presentará la interfaz gráfica de Thunderbird, en la cual se ingresa a uno de los usuarios registrados para realizar el envío de un mensaje a otro usuario existente. Adicionalmente, al iniciar sesión otro usuario con los pasos anteriormente descritos, en la Figura 3.102 se observa cómo se abre el editor de correo en el que se envía un correo desde el usuario Jenny hasta el usuario Tesis.

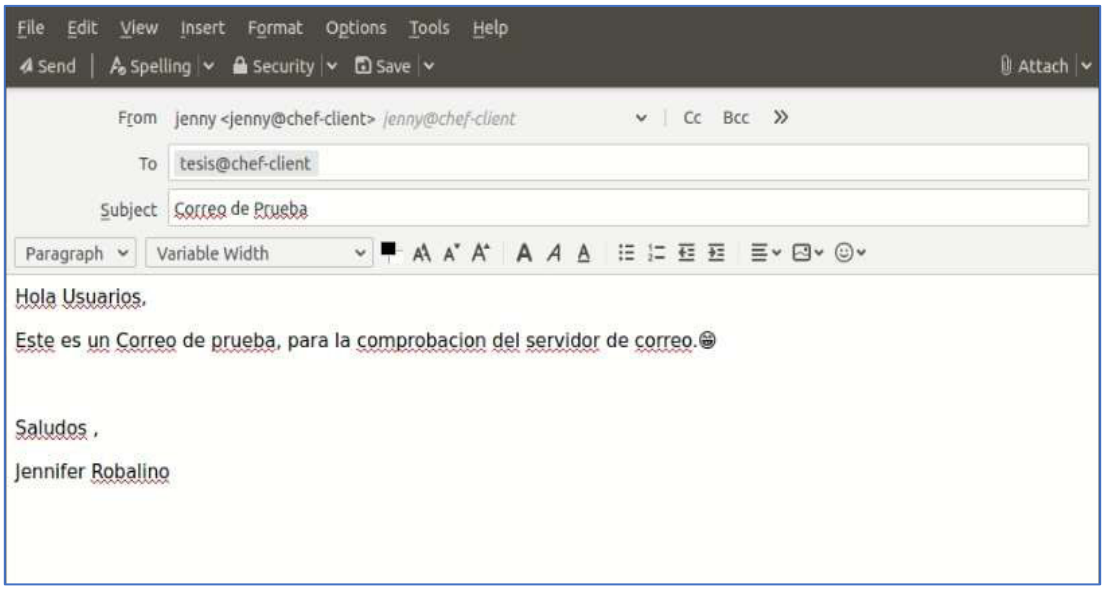


**Figura 3.102** Correo de prueba desde usuario 1 hasta usuario 2

Enviado el mensaje, en la bandeja de entrada del usuario destino se podrá ver el mensaje anteriormente escrito por el remitente, como se indica en la Figura 3.103, donde se pueden apreciar también parámetros como el asunto, el usuario receptor y la hora de recepción del mensaje.

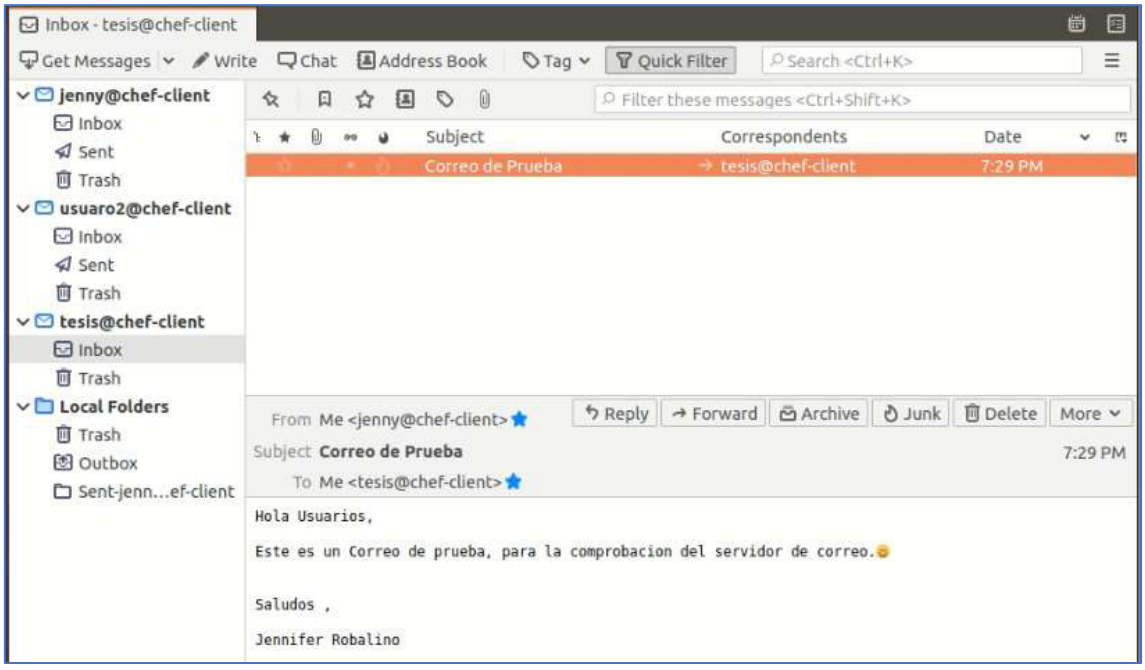


**Figura 3.103** Mensaje de prueba recibido por el destinatario

# 4 CONCLUSIONES Y RECOMENDACIONES

## 4.1 Conclusiones

- El *software* Chef permite la optimización de las operaciones del departamento de telecomunicaciones, dentro de una organización, por lo que para lograrlo se utilizó cuatro componentes principales siendo el más relevante de ellos Chef *Workstation*. La administración de este nodo central permitió brindar a los nodos clientes los recursos necesarios para la ejecución de los servicios, los cuales requieren de configuraciones específicas en cada una de las recetas.
- Uno de los principales inconvenientes al momento de la definición de recetas fue el manejo del lenguaje de programación Ruby, adicionalmente se debe tener en consideración que se debe utilizar la instalación completa de la versión más actual del paquete de Ruby para que la ejecución sea la correcta.
- Es importante tener en consideración que se debe configurar un canal seguro entre el servidor, la estación de trabajo y el nodo cliente, para ello es necesario usar el intercambio de claves privadas generadas mediante ssh con el objetivo de que los datos que viajan entre los equipos se vean cifradas manteniendo de esa manera la confidencialidad de la información.
- Uno de los principales aportes del *software* Chef es brindar un entorno de configuración más sencillo para los administradores de la red, esto se realiza por medio de la instalación de Chef Manage dentro de Chef *server*. Dentro de la cual se detallan aspectos relevantes como la dirección de los nodos, el ambiente de ejecución y el nombre de dominio completo.
- La configuración de los ficheros que conforman cada una de las recetas elaboradas debe ser perfectamente realizada en sus roles y recursos para que el archivo principal de ejecución default.rb pueda hacer uso de estos recursos adecuadamente.
- Para que el servicio DNS funcione adecuadamente, los archivos que contienen la configuración de zona directa y recursiva deben ser descritos correctamente, ya que se detalla la información sobre la traducción de nombres de dominio a IP o viceversa.
- Debido a las limitaciones de recursos del servidor provisto por la Escuela Politécnica Nacional se tuvo que crear en un mismo cliente varios adaptadores de red para realizar las pruebas de funcionamiento y de este modo estos cumplen las funciones de cliente DHCP.

- El principal inconveniente en la implementación del servicio de correo fue las versiones de los *software* disponibles que permiten leer los paquetes recibidos desde el MUA. El primer programa utilizado llamado Squirrelmail no fue compatible con la versión de Ubuntu 18.04 manejada en este proyecto por lo que se tuvo que utilizar el *software* de correo por defecto del sistema operativo denominado Thunderbird.

## 4.2 Recomendaciones

- Se recomienda previamente a utilizar el *software* Chef realizar los cursos gratuitos disponibles en la plataforma de chef.io, de este modo se tendrá un mejor control sobre las herramientas disponibles en este *software*, logrando optimizar tiempo en la implementación.
- Debido a que actualmente no se dispone de documentación actualizada suficiente del *software* Chef debido a que las últimas versiones de este fueron lanzadas recientemente. Es recomendable que se utilice versiones anteriores como la versión 14 la cual posee un mejor control de errores evitando así que los servicios a implementar se vean afectados.
- Es recomendable que para la implementación de este *software* se utilice una de las herramientas disponibles en la web siendo una de las más relevantes AWS (Amazon Web Services) la cual permite utilizar los recursos disponibles en la nube optimizando así todos los servicios a proveer dentro de la organización.
- El *software* Chef dispone de unas licencias gratuitas disponibles para los usuarios en un periodo de prueba, una de las principales limitaciones al momento de utilizar estas licencias gratuitas es el uso de herramientas específicas las cuales permiten solo añadir una cierta cantidad de servicios, así como la cantidad de usuarios disponibles para añadir a Chef *Server*.
- Previo a la implementación de este *software* se debe considerar los recursos necesarios que posea la organización para evitar latencias en el envío de la información y de igual manera saber qué versión y qué herramientas tendrá disponibles el administrador de la red.
- Se recomienda adquirir las recetas desde la página oficial de Chef denominada Supermarket dentro de la cual se especificarán todas las recetas, así como sus actualizaciones y las organizaciones que las manejan, dando una mejor perspectiva de sus capacidades y un mejor panorama para la implementación de las mismas.

# 5 REFERENCIAS BIBLIOGRAFICAS


[1] «Chef Progress,» An Overview of Chef InSpec, 22 Febrero 2021. [En línea]. Available: https://docs.chef.io/inspec/. [Último acceso: 27 Julio 2021].

[2] L. E. V. Alzate, «ESPECIALIZACION EN GERENCIA INTEGRAL DE PROYECTOS,» Diciembre 2014. [En línea]. Available: https://n9.cl/e7q0e. [Último acceso: 15 Julio 2021].

[3] AtuestaDaniel, «Slideshare,» Herramientas DevOps, 21 Abril 2016. [En línea]. Available: https://es.slideshare.net/AtuestaDaniel/herramientas-devops. [Último acceso: 15 Julio 2021].

[4] «Red Hat,» DevSecOps y la seguridad de DevOps, 2021. [En línea]. Available: https://www.redhat.com/es/topics/devops/what-is-devsecops. [Último acceso: 10 Julio 2021].

[5] A. S. V. N. C. LEODAN, «UNIVERSIDAD DE GUAYAQUIL,» 2016. [En línea]. Available: https://n9.cl/xyj3c. [Último acceso: 15 Julio 2021].

[6] Yuly Andrea Beltran Ruiz, Implementación servicios de infraestructura IT, Escuela de Ciencias Básicas Tecnología e Ingeniería - ECBTI, Universidad Nacional Abierta y a Distancia, 2018.

[7] «CHEF,» 2020. [En línea]. Available: https://www.chef.io/webinars. [Último acceso: 14 Septiembre 2021].

[8] «Linke Cloud Technology for Enterprises,» Linke IT, [En línea]. Available: https://www.linkeit.com/es/blog/que-es-*software*-chef. [Último acceso: 15 Julio 2021].

[9] M. W. Navin Sabharwal, Automation through Chef Opscode, Apress, 2014.

[10] «Chef Progress,» Chef Infra *Server* Overview, 17 Marzo 2021. [En línea]. Available: https://docs.chef.io/*server*/. [Último acceso: 10 Julio 2021].

[11] «Chef Progress,» Chef Infra Client Overview, 28 Abril 2021. [En línea]. Available: https://docs.chef.io/chef_client_overview/. [Último acceso: 25 Julio 2021].

[12] «Chef Progress,» About Chef *Workstation*, 30 Abril 2021. [En línea]. Available: https://docs.chef.io/*workstation*/. [Último acceso: 25 Junio 2021].

[13] «Indellient,» 6 Agosto 2019. [En línea]. Available: https://docs.chef.io/habitat/.

[14] L. Chef, «Chef Progress,» Gestión de la infraestructura de Chef, 2021. [En línea]. Available: https://learn.chef.io/courses/course-v1:chef+Chef101+Perpetual/courseware/d556513fb56a480db4aaa41c88d7aea6/422a05eb1a3c44ff81b70e05680954cc/. [Último acceso: 16 Julio 2021].

[15] C. -. Overview., «Tutorialspoint,» 2021. [En línea]. Available: https://www.tutorialspoint.com/chef/chef_overview.htm. [Último acceso: 14 Septiembre 2021].

[16] «Chef Progress,» Acerca de los libros de cocina, 12 Marzo 2021. [En línea]. Available: https://docs.chef.io/*cookbook*s/. [Último acceso: 28 Junio 2021].

[17] «Chef Progress,» Acerca de *Knife*, 12 Marzo 2021. [En línea]. Available: https://docs.chef.io/*workstation*/*knife*/. [Último acceso: 24 Julio 2021].

[18] «Ruby,» Ruby es.., 2021. [En línea]. Available: https://www.ruby-lang.org/es/. [Último acceso: 27 Julio 2021].

[19] «VMware, Inc,» 2021. [En línea]. Available: https://www.vmware.com/co/products/*workstation*-pro.html. [Último acceso: 15 Juio 2021].

[20] «IBM Docs,» 2021. [En línea]. Available: https://www.ibm.com/docs/es/i/7.2?topic=concepts-bind-9-features. [Último acceso: 13 Septiembre 2021].

[21] «DigitalOcean, LLC,» Cómo Configurar BIND Como Servidor DNS De Red Privada En Ubuntu 18.04, 2021. [En línea]. Available: https://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-private-network-dns-*server*-on-ubuntu-18-04-es.. [Último acceso: 13 Septiembre 2021].

[22] serviciosgs, «Ejercicio: Instalación y configuración del servidor dhcp en linux,» 2017. [En línea]. Available: https://serviciosgs.readthedocs.io/es/latest/dhcp/ejercicio1.html. [Último acceso: 13 Septiembre 2021].

[23] P. &. Python, «Correo electrónico con Postfix, Dovecot y Thunderbird en Ubuntu 20.04.,» 2021. [En línea]. Available: https://elpuig.xeill.net/Members/vcarceler/articulos/correo-electronico-con-postfix-dovecot-y-thunderbird-en-ubuntu-20.04. [Último acceso: 13 Septiembre 2021].

[24] Mailjet, «¿Qué Es Un Servidor SMTP? Qué Significa Y Cómo Usarlo Para Enviar Email,» Email Marketing *Software*, 2020. [En línea]. Available: https://es.mailjet.com/blog/news/servidor-smtp/.. [Último acceso: 13 Septiembre 2021].

[25] Tecnologia-Facil.Com., «El Protocolo IMAP - Tecnología Fácil,» 2021. [En línea]. Available: https://tecnologia-facil.com/que-es/protocolo-imap/#%C2%BFQue_es_el_protocoloIMAP. [Último acceso: 13 Septiembre 2021].

# ANEXOS

# ANEXO 1: CERTIFICADO DE FUNCIONAMIENTO

**ESCUELA POLITECNICA NACIONAL**
Campus Politécnico "J. Rubén Orellana R

Quito, 14 de septiembre de 2021

# CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN

Yo, *Fernando Vinicio Becerra Camacho,* docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de titulación, certifico que he constatado el correcto funcionamiento de la simulación de los servidores DNS, DHCP y correo mediante el *software* chef implementados por la estudiante Jennifer Alexandra Robalino Alcívar.

**DIRECTOR**

Ing. Fernando Vinicio Becerra Camacho., Msc.

Ladrón de Guevara E11-253, Escuela de Formación de Tecnólogos, Oficina 28. EXT: 2729
email: fernando.becerrac@epn.edu.ec Quito-Ecuador

# ANEXO 2: COMANDOS DE EDICIÓN VI

# EDITOR VI

El editor vi es un editor de texto que maneja en memoria el texto entero de un archivo. Es el editor clásico de UNIX (se encuentra en todas las versiones). Puede usarse en cualquier tipo de terminal con un mínimo de teclas, lo cual lo hace difícil de usar al enfrentarse por primera vez al mismo.

## MODOS DE VI:

Existen tres modos o estados de vi:

- **Modo comando:** este es el modo en el que se encuentra el editor cada vez que se inicia. Las teclas ejecutan acciones (comandos) que permiten mover el cursor, ejecutar comandos de edición de texto, salir de vi, guardar cambios, etc.
- **Modo inserción o texto:** este es el modo que se usa para insertar el texto. Existen varios comandos que se pueden utilizar para ingresar a este modo.
- **Modo línea o ex:** se escriben comandos en la última línea al final de la pantalla.

## INICIO DE VI:

**`vi`**

Abre la ventana de edición sin abrir ningún archivo.

**`vi`** *`archivo1`*

Edita el archivo *archivo1* si ya existe, de lo contrario, lo crea. Evidentemente se debe indicar el camino (path) que conduce al archivo (si existe) o el camino que conduce al directorio donde se desea crear el archivo (si este no existe).

## MODO COMANDO:

El editor vi, como todo UNIX, diferencia mayúsculas de minúsculas. A continuación se comentan algunos comandos útiles en el manejo del editor.

**Movimiento del cursor:**

| Comando (teclas) | Acción |
|---|---|
| `Flechas` | Mover en la dirección de la flecha |
| `h` | Mover hacia la izquierda |
| `l` | Mover hacia la derecha |
| `k` | Mover hacia arriba |
| `j` | Mover hacia abajo |
| `1G` | Lleva el cursor hasta el comienzo del archivo |
| `G` | Lleva el cursor hasta el final del archivo |

**Cambio de modo comando a texto:**

| Comando | Acción |
|---|---|
| `i` | Inserta texto a la izquierda del cursor |
| `a` | Inserta texto a la derecha del cursor |
| `A` | Inserta texto al final de la línea donde se encuentra el cursor |
| `I` | Inserta texto al comienzo de la línea donde se encuentra el cursor |
| `o` | Abre una línea debajo de la actual |
| `O` | Abre una línea encima de la actual |

**Borrar texto:**

| Comando | Acción |
|---|---|
| `x` | Borra el carácter bajo el cursor |
| `dd` | Borra la línea donde se encuentra el cursor |
| `ndd` | Borra las próximas n líneas |
| `D` | Borra desde donde se encuentra el cursor hasta el final de la línea |
| `dw` | Borra desde donde se encuentra el cursor hasta el final de una palabra |

Es importante destacar que todo lo que se borra queda almacenado en un buffer (área temporal de memoria), de modo que si se borró algo por error, puede volver a escribirse (si se hace antes de realizar otros cambios, es decir, inmediatamente luego de eliminar el texto por error. Esto se hace simplemente ejecutando el comando **p**.

**Cortar y pegar:**

Esto implica mover partes del archivo de un lugar a otro del mismo. Para esto se debe:

- Cortar el texto que se desea mover utilizando alguno de los comandos usados para borrar texto.
- Mover el cursor (con alguno de los comandos utilizados para desplazar el cursor en el texto) hasta el lugar donde se desee pegar el texto.
- Pegar el texto con el comando **p**.

**Copiar y pegar:**

Esta operación difiere de la anterior. En este caso lo que se hace es repetir partes del texto en otro lugar del archivo. Para esto se debe:

- Utilizar el comando **yy**, cuya función es copiar la línea donde se encuentra situado el cursor.
- Mover el cursor (con alguno de los comandos utilizados para desplazar el cursor en el texto) hasta el lugar donde se desee pegar el texto.
- Pegar el texto con el comando **p**.

**Deshacer cambios:**

Se puede deshacer el último cambio realizado, utilizando el comando `u`.

## MODO TEXTO:

En este modo se ingresa el texto deseado. Para pasar de modo texto a modo comando simplemente se debe apretar la tecla ESC.

## MODO LÍNEA:

Para ingresar al modo línea desde el modo comando, se debe utilizar alguna de las siguientes teclas:

`/`
`?`
`:`

Para volver al modo comando desde el modo última línea, se debe apretar la tecla ENTER (al finalizar el comando) o la tecla ESC (que interrumpe el comando).

**Buscar texto:**

| Comando | Acción |
|---|---|
| `/texto` | Busca hacia adelante la cadena de caracteres "texto" |
| `?texto` | Busca hacia atrás la cadena de caracteres "texto" |

**Salir de vi, salvar, no salvar cambios, etc.:**

| Comando | Acción |
|---|---|
| `:q` | Salir si no hubo cambios |
| `:q!` | Salir sin guardar cambios |
| `:w` | Guardar cambios |
| `:w archivo1` | Guardar cambios en archivo1 |
| `:wq` | Guardar cambios y salir |

# ANEXO 3: LISTA DE COMANDOS CHEF *SERVER*

```
1.    sudo nano /etc/hosts
2.     hostname -f
3.     sudo apt update
4.     wget https://packages.chef.io/files/stable/chef-
   server/13.0.17/ubuntu/18.04/chef-server-core_13.0.17-
   1_amd64.deb
5.      sudo dpkg -i chef-server-core_*.deb
6.      sudo chef-server-ctl status
7.      sudo chef-server-ctl reconfigure
8.      sudo apt install ntp
9.      systemctl start ntp
10.  systemctl enable ntp
11.  timedatectl
12.  sudo chef-server-ctl install chef-manage
13.  sudo chef-server-ctl reconfigure
14.  sudo chef-manage-ctl reconfigure
15.  sudo chef --version
16.  mkdir .chef
17.  sudo chef-server-ctl user-create chefadmin chef tesis
   chefadmin@tesis.com '0901971290' --filename
   ~/.chef/chefadmin.pem
18.  sudo chef-server-ctl user-list
19.  sudo chef-server-ctl org-create chef-org "tesis chef
   infrastructure" --association_user chefadmin --filename
   ~/.chef/chef-org.pem
20.  sudo chef-server-ctl org-list
21.  sudo chef-server-ctl install opscode-reporting
22.  sudo chef-server-ctl reconfigure
23.  sudo opscode-reporting-ctl reconfigure
24.  sudo chef-server-ctl install opscode-push-jobs-server
25.  sudo chef-server-ctl reconfigure
26.  sudo opscode-push-jobs-server-ctl reconfigure
27.  test opscode-reporting-ctl
```

# ANEXO 4: LISTA DE COMANDOS CHEF *WORKSTATION*

```
1.  sudo apt install net-tools
2.  ifconfig
3.  sudo nano /etc/hosts
4.  hostname -f
5.  sudo apt update
6.  sudo nano /etc/hosts
7.  wget https://packages.chef.io/files/stable/chef-
    workstation/0.2.43/ubuntu/18.04/chef-workstation_0.2.43-
    1_amd64.deb
8.  sudo dpkg -i chef-workstation_*.deb
9.  chef generate repo chef-repo
10. mkdir ~/chef-repo/.chef
11. cd chef-repo
12. ssh-keygen -b 4096
13. sudo ssh-copy-id
14. sudo ssh-copy-id tesis@192.168.247.209
15. scp tesis@192.168.247.209: ~/.chef/*.pem ~/chef-
    repo/.chef/
16. sudo scp tesis@192.168.247.209: ~/.chef/*.pem ~/chef-
    repo/.chef/
17. scp tesis@192.168.247.209:~/.chef/*.pem ~/chef-repo/.chef/
18. sudo ssh-copy-id tesis@192.168.247.209
19. ssh-copy-id tesis@192.168.247.209
20. scp tesis@192.168.247.209:~/.chef/*.pem ~/chef-repo/.chef/
21. ls ~/chef-repo/.chef
22. sudo apt-get install git
23. git config  --global  user.name jennyssale
24. git config  --global  user.email
    jennifer_robalino1997@hotmail.com
25. echo ".chef" > ~/chef-repo/.gitignore
26. cd ~/chef-repo
27. sudo git init
28. sudo git add .
29. sudo git commit -m "initial
30. git status
31. chef generate cookbook libro_1
32. sudo nano sudo ~/chef-repo/.chef/config.rb
33. sudo nano  ~/chef-repo/.chef/config.rb
34. chef generate app chef-repo
35. sudo gedit ~/chef-repo/.chef/config.rb
36. sudo nano  ~/chef-repo/.chef/config.rb
37. cd chef-repo/cookbooks/
38. rm -r libro_1/
39. chef generate cookbook cookbooks/libro_1
40. vim cookbooks/libro_1/metadata.rb
41. sudo vim cookbooks/libro_1/metadata.rb
42. sudo apt install vim
43. sudo vim cookbooks/libro_1/metadata.rb
44. cd chef-repo/
45. knife ssl fetch
```

```
46.  knife client list
47.  knife bootstrap 192.168.247.211 -x tesis -P 12345 --node-
   name chefadmin
48.  chef --version
49.  ssh tesis@192.168.247.209
50.  ssh tesis@192.168.247.210
51.  sudo apt-get install openssh-server
52.  sudo apt-get install openssh-server
53.  sudo ufw allow 22
54.  sudo service ssh status
55.  sudo apt-get install openssh-server
56.  knife supermarket download chef-client
57.  cd chef-repo/cookbooks/
58.  knife supermarket download cron
59.  knife supermarket download logrotate
60.  sudo apt-get install openssh-server
61.  cd chef-repo/cookbooks/
62.  cd cookbooks/
63.  knife supermarket install logrotate
64.  knife supermarket install chef-client
65.  tar -xvzf chef-client-12.3.4.tar.gz
66.  tar -xvzf cron-7.0.2.tar.gz cron/
67.  tar -xvzf logrotate-3.0.3.tar.gz
68.  ls
69.  $ knife upload cookbook cron
70.   ls
71.   cd
72.  sudo apt-get update
73.  sudo apt-get install openssh-server
74.  sudo systemctl start sshd.service
75.  sudo systemctl stop sshd.service
76.  sudo systemctl restart sshd.service
77.  sudo systemctl status sshd.service
78.  shh tesis@192.168.247.211
79.  sudo shh tesis@192.168.247.211
80.  shh tesis@192.168.247.209
81.  sudo systemctl status sshd
82.  ssh tesis@192.168.247.209
83.  ssh tesis@192.168.247.211
84.   sudo systemctl enable sshd.service
85.  sudo nano /etc/ssh/sshd_config
86.  sudo systemctl restart sshd.service
87.  ssh 192.168.247.211
88.  sudo apt-get install openssh-client
89.  ssh tesis@192.168.247.211
90.  ssh tesis@192.168.247.209
91.  cd chef-repo/cookbooks/
92.  knife bootstrap 192.168.247.211 -x tesis -P 12345 --node-
   name chefadmin
```

```
93.   knife bootstrap 192.168.247.211 --ssh-user tesis --sudo --
 identity-file ~/.ssh/id_rsa.pub --node-name chefadmin
94.   ls
 95.  cd ..
 96.  knife upload cookbook cookbooks/chef-client
 97.  cd cookbooks/
 98.  ls
 99.  cd cron
 100. ls
 101. tree
 102. sudo apt install tree
 103. tree
 104. cd ..
 105. tree
 106. knife upload cookbook chef-client
 107. knife upload cookbook logrotate
 108. cd
 109. sudo  vim cookbooks/libro_1/metadata.rb
 110. sudo vi chef-repo/cookbooks/libro_1/metadata.rb
 111. sudo vi chef-repo/cookbooks/libro_1/recipes/default.rb
 112. sudo apt install ruby-full
 113. sudo vi chef-repo/cookbooks/libro_1/recipes/default.rb
 114. git add cookbooks/libro_1
 115. cd chef-repo/
 116. git add cookbooks/libro_1
 117. cd .git/
 118. git add cookbooks/libro_1
 119. cd ..
 120. cd chef-repo/
 121. cd cookbooks/
 122. knife upload cookbook libro_1
 123. cd
 124. cd chef-repo/
 125. knife node list
 126. knife node show chefadmin
 127. ssh tesis@192.168.247.211
 128. cd cookbooks/
 129. knife cookbook detele logrotate
 130. knife cookbook detele logrotate 3.0.3
 131. knife cookbook detele logrotate
 132. cd ..
 133. knife cookbook detele logrotate
 134. cd cookbooks/
 135. knife cookbook delete logrotate
 136. ls
 137. rm -r logrotate
 138. rm -r logrotate-3.0.3.tar.gz
 139. knife supermarket install logrotate 2.2.0
 140. knife supermarket download logrotate 2.2.0
 141. tar -xvzf logrotate-2.2.0.tar.gz
```

```
142. ls
143. cd logrotate/
144. tree
145. cd ..
146. cd
147. ssh tesis@192.168.247.211
148. cd chef-repo/cookbooks
149. chef generate cookbook bind9
150. chef generate cookbook dhcp
151. chef generate cookbook postfix
152. chef generate cookbook dovecot
153. *
154. cd bind9/
155. tree
156. sudo vi recipes/default.rb
157. chef generate
158. chef generate template named.conf.local
159. chef generate template mapadirecto.chef-client
160. tre
161. tree
162. vi templates/mapadirecto.chef-client.erb
163. sudo vi templates/named.conf.local.erb
164. sudo vi template
165. chef generate template inverso.192.168.247
166. tree
167. sudo vi templates/inverso.192.168.247.erb
168. sudo vi templates/named.conf.local.erb
169. sudo vi templates/inverso.192.168.247.erb
170. sudo vi metadata.rb
171. sudo vi recipes/default.rb
172. cd ..
173. knife cookbook upload bind9
174. chef exec ruby -c chef-
    repo/cookbooks/bind9/recipes/default.rb
175. cd
176. chef exec ruby -c chef-
    repo/cookbooks/bind9/recipes/default.rb
177. sudo vi chef-repo/cookbooks/bind9/recipes/default.rb
178. chef exec ruby -c chef-
    repo/cookbooks/bind9/recipes/default.rb
179. sudo vi chef-repo/cookbooks/bind9/recipes/default.rb
180. chef exec ruby -c chef-
    repo/cookbooks/bind9/recipes/default.rb
181. tree
182. chef exec ruby -c chef-
    repo/cookbooks/bind9/templates/named.conf.local.erb
183. sudo vi  chef-repo/cookbooks/bind9/recipes/default.rb
184. sudo vi  chef-
    repo/cookbooks/bind9/templates/named.conf.local.erb
```

```
185. chef exec ruby -c chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
186. sudo vi  chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
187. chef exec ruby -c chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
188. sudo vi  chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
189. chef exec ruby -c chef-repo/cookbooks/bind9/templates/
190. cd chef-repo/cookbooks/
191. cd
192. chef exec ruby -c chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
193. sudo vi  chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
194. chef exec ruby -c chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
195. sudo vi  chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
196. chef exec ruby -c chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
197. sudo vi  chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
198. chef exec ruby -c chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
199. chef exec ruby -c chef-
   repo/cookbooks/bind9/templates/mapadirecto.chef-client.erb
200. sudo vi chef-
   repo/cookbooks/bind9/templates/mapadirecto.chef-client.erb
201. chef exec ruby -c chef-
   repo/cookbooks/bind9/templates/mapadirecto.chef-client.erb
202. repo/cookbooks/bind9/templates/inverso.192.168.247.erb
203. sudo vi chef-
   repo/cookbooks/bind9/templates/inverso.192.168.247.erb
204. sudo vi chef-
   repo/cookbooks/bind9/templates/mapadirecto.chef-client.erb
205. sudo vi chef-repo/cookbooks/bind9/recipes/default.rb
206. sudo vi chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
207. cd chef-repo/cookbooks/
208. knife cookbook upload bind9
209. cd
210. sudo vi chef-
   repo/cookbooks/bind9/templates/named.conf.local.erb
211. cd chef-repo/cookbooks/
212. ls
213. cd dhcp/
214. tree
215. sudo vi recipes/default.rb
216. chef generate template isc-dhcp-server
```

```
217. chef generate template dhcpd.conf
218. knife cookbook upload dhcp
219. cd dhcp/
220. cd ..
221. knife cookbook upload dhcp
222. chef exec ruby -c chef-
   repo/cookbooks/dhcp/recipes/default.rb
223. cd
224. chef exec ruby -c chef-
   repo/cookbooks/dhcp/recipes/default.rb
225. sudo vi chef-repo/cookbooks/dhcp/recipes/default.rb
226. chef exec ruby -c chef-
   repo/cookbooks/dhcp/recipes/default.rb
227. cd chef-repo/cookbooks/
228. knife cookbook upload dhcp
229. postfix/
230. cd postfix/
231. sudo vi recipes/default.rb
232. cd
233. chef exec ruby -c chef-
   repo/cookbooks/postfix/recipes/default.rb
234. cd chef-repo/cookbooks/
235. knife cookbook upload postfix
236. cd dhcp/

237. tree
238. cd ..
239. cd postfix/
240. sudo vi recipes/default.rb
241. cd
242. chef exec ruby -c chef-
   repo/cookbooks/postfix/recipes/default.rb
243. cd chef-repo/cookbooks/
244. knife cookbook upload postfix
245. sudo vi dovecot/recipes/default.rb
246. cd
247. chef exec ruby -c chef-
   repo/cookbooks/dovecot/recipes/default.rb
248. cd chef-repo/cookbooks/
249. knife cookbook upload dovecot
250. sudo vi dovecot/recipes/default.rb
251. cd
252. cd chef-repo/cookbooks/dovecot/
253. chef generate template 10-auth.conf
254. chef generate template 10-mail.conf
255. knife cookbook upload dovecot
256. cd chef-repo/cookbooks/dovecot/
257. sudo vi dovecot/recipes/default.rb
258. cd ..
259. sudo vi dovecot/recipes/default.rb
```

```
260. knife cookbook upload dovecot
261. sudo vi dovecot/recipes/default.rb
262. sudo vi dovecot/templates/10-mail.conf.erb
263. sudo vi dovecot/templates/10-auth.conf.erb
264. cd dovecot/
265. tree
266. sudo vi /recipes/default.rb
267. sudo vi recipes/default.rb
268. cd
269. cd chef-repo/cookbooks/
270. cd postfix/
271. tree
272. sudo vi recipes/default.rb
273. chef generate template main.cf
274. tree
275. sudo vi templates/main.cf.erb
276. sudo vi recipes/default.rb
277. cd ..
278. tree
279. sudo vi templates/dhcpd.conf.erb
280. sudo vi templates/isc-dhcp-server.erb
281. cd ..
282. cd bind9
283. tree
284. sudo vi templates/inverso.192.168.247.erb
285. sudo vi templates/named.conf.local.erb
286. sudo vi templates/mapadirecto.chef-client.erb
287. sudo vi templates/inverso.192.168.247.erb
```

# ANEXO 5: LISTA DE COMANDOS CHEF CLIENT

```
1.    sudo apt install net-tools
2.    ifconfig
3.    sudo nano /etc/hosts
4.    hostname -f
5.    sudo apt update
6.    sudo apt-get install opssh
7.    sudo service ssh status
8.    mkdir .chef
9.    .chef
10.   cd .chef
11.   scp tesis@192.168.247.209:~/.chef/*.pem ~/.chef/
12.   sudo ssh-copy-id  tesis@192.168.247.209
13.   ls
14.   cd
15.   sudo apt-get install ssh
16.   sudo apt install ssh
17.   ssh tesis@192.168.247.210
18.   sudo service ssh status
19.   sudo systemctl start sshd.service
20.   sudo systemctl status sshd.service
21.   ssh tesis@192.168.247.209
22.   sudo apt install ruby-full
23.   ifconfig
24.   ping 192.168.247.211
25.   ping 192.168.247.218
26.   sudo chef-client
27.   sudo chef-client
28.   sudo /etc/init.d/bind9 restart
29.   sudo service bind9 status
30.   ping -c 4 192.168.247.211
31.   ping -c 4 server
32.   nslookup server
33.   nslookup 192.168.247.211
34.   nslookup correo.chef-client
35.   pin chef-workstation
36.   nslookup client.chef-client
37.   nslookup servidor.chef-client
38.   sudo chef-client
39.   sudo nano /etc/network/interfaces
40.   sudo systemctl start isc-dhcp-server
41.   sudo systemctl enable isc-dhcp-server
42.   sudo systemctl status isc-dhcp-server
43.   ip a
44.   ifconfig
45.   sudo ifdown ens40
46.   sudo ifup ens40
47.   ip a
48.   dhcp-lease-list
49.   sudo chef-client
50.   sudo adduser jenny
51.   sudo adduser usuario2
52.   sudo ls -l /home/
53.   sudo thunderbird
```

# ANEXO 6: VIDEO DE CONCEPTOS GENERALES *SOFTWARE* CHEF

# ANEXO 7: VIDEO INSTALACIÓN SERVICIOS DE CHEF

# ANEXO 8: VIDEO INSTALACIÓN SERVIDORES DNS, DHCP Y CORREO

# ANEXO 9: VIDEO VERIFICACIÓN DE SERVIDORES