

Universidad Tecnológica de San Luis Potosí

Memoria Técnica: Instalación de un clúster de Kubernetes con 3 nodos

Alumna: Lizeth Anahí Arenas Ramírez

Grupo: IRC 10.2

31 de mayo de 2024

Prof. Ing. Omar Cruz Gutiérrez

**CONTENIDO**

[**1.** **OBJETIVO** 4](#_Toc168129704)

[**2.** **ALCANCE** 4](#_Toc168129705)

[**3.** **REQUERIMENTOS** 4](#_Toc168129706)

[**3.1.** **Requisitos de Hardware** 4](#_Toc168129707)

[**3.2.** **Requisitos de Software** 5](#_Toc168129708)

[**4.** **PASOS PARA LA INSTALACIÓN** 5](#_Toc168129709)

[**4.1.** **Actualización del sistema e instalación.** 5](#_Toc168129710)

[**4.2.** **Configuración Ansible** 7](#_Toc168129711)

[**4.3.** **Instalación de Kubernetes (k8s)** 11](#_Toc168129712)

[**4.4.** **Playbook para instalar Kubernetes** 12](#_Toc168129713)

[**4.5.** **Revisión del servicio SSH** 13](#_Toc168129714)

[**4.6.** **Ejecución de Playbook** 14](#_Toc168129715)

[**4.6.1.** **Errores al ejecutar Playbook de Ansible** 16](#_Toc168129716)

[**5.** **REVISIÓN DE SERVICIOS** 19](#_Toc168129717)

**ÍNDICE DE IMÁGENES**

[Fig. 1-Actualización de sistema 6](#_Toc168129718)

[Fig. 2-Instalación de herramientas y utilidades 6](#_Toc168129719)

[Fig. 3-Instalación de repositorio epel 6](#_Toc168129720)

[Fig. 4-Instalación ansible 7](#_Toc168129721)

[Fig. 5-Versión Ansible 7](#_Toc168129722)

[Fig. 6-Configuración de hosts 8](#_Toc168129723)

[Fig. 7-Archivo de hosts del sistema 8](#_Toc168129724)

[Fig. 8-Generar clave SSH 9](#_Toc168129725)

[Fig. 9-Clave ssh de cada nodo agregada al master 10](#_Toc168129726)

[Fig. 10-Cambio de nombre de nodo master 10](#_Toc168129727)

[Fig. 11-Clonación de repositorio de Kubernetes 11](#_Toc168129728)

[Fig. 12-Clústers definidos en hosts 11](#_Toc168129729)

[Fig. 13-Contenido del archivo 'k8s-prep-yml' 13](#_Toc168129730)

[Fig. 14-Comandos aplicados a nodo master 14](#_Toc168129731)

[Fig. 15-Conexión SHH de master hacia nodo1 15](#_Toc168129732)

[Fig. 16-Verificando si es correcta la sintaxis 15](#_Toc168129733)

[Fig. 17-Inicio de agente y clave SSH 15](#_Toc168129734)

[Fig. 18-Ejecución de Playbook 16](#_Toc168129735)

[Fig. 19-Ejecución exitosa de los 3 nodos 16](#_Toc168129736)

[Fig. 20-Error de módulo ansible.posix.selinux 16](#_Toc168129737)

[Fig. 21-Búsqueda en directorio de archivo pre\_setup.yml 17](#_Toc168129738)

[Fig. 22-Archivo comentado en las 4 primeras líneas 17](#_Toc168129739)

[Fig. 23-Error de community.general 18](#_Toc168129740)

[Fig. 24-Instalación de módulo community.general 18](#_Toc168129741)

[Fig. 33-Servicio crio activo correctamente 19](#_Toc168129742)

[Fig. 35-Firewall desactivado 19](#_Toc168129743)

1. **OBJETIVO**

El objetivo de este proyecto es implementar un clúster de Kubernetes compuesto por tres nodos, automatizando el proceso de instalación y configuración mediante Ansible para garantizar una implementación eficiente, consistente y reproducible. Esta automatización busca facilitar la gestión y el escalado del clúster, asegurando alta disponibilidad y optimizando el rendimiento de las aplicaciones desplegadas en contenedores dentro del entorno de Kubernetes.

1. **ALCANCE**

El alcance incluye la preparación de los nodos con los requisitos necesarios, la instalación de Docker y Kubernetes, la inicialización del nodo maestro, y la integración de los nodos de trabajo al clúster. Se cubren las configuraciones de red y las verificaciones post-instalación para asegurar el correcto funcionamiento y alta disponibilidad del clúster. Esta guía está orientada a administradores de sistemas que buscan una implementación eficiente y escalable de Kubernetes.

1. **REQUERIMENTOS**
   1. **Requisitos de Hardware**

Nodo Master

* CPU: 2 vCPUs (mínimo)
* Memoria RAM: 2 GB (mínimo)
* Almacenamiento: 20 GB de espacio en disco
* Conectividad: Acceso a red y a Internet

Nodos Trabajadores (2)

* CPU: 2 vCPUs (mínimo) cada uno
* Memoria RAM: 2 GB (mínimo) cada uno
* Almacenamiento: 20 GB de espacio en disco cada uno
* Conectividad: Acceso a red y a Internet
  1. **Requisitos de Software**

Sistema Operativo

* Distribución: CentOS 7 (completamente actualizado con los últimos parches de seguridad)

Paquetes y Dependencias

* Ansible: Instalado en la máquina de administración (versión mínima recomendada: 2.9)
* Docker: Instalado en todos los nodos (versión recomendada: 19.03 o superior)
* Kubeadm, Kubelet, Kubectl: Instalados en todos los nodos (versión recomendada: 1.20 o superior)

1. **PASOS PARA LA INSTALACIÓN**

Los siguientes pasos detallan el proceso de instalación y configuración de un clúster de Kubernetes con tres nodos utilizando Ansible. Este procedimiento incluye la preparación de los nodos, la configuración de los servicios necesarios y la aplicación de las configuraciones específicas de Kubernetes, todo mediante un playbook de Ansible. Este enfoque automatiza y simplifica la implementación, asegurando consistencia y reduciendo errores en la configuración de los nodos maestro y trabajadores dentro del clúster.

* 1. **Actualización del sistema e instalación.**

Para iniciar el proceso de instalación, se actualizan todos los paquetes instalados en el sistema a las versiones más recientes.

yum -y update

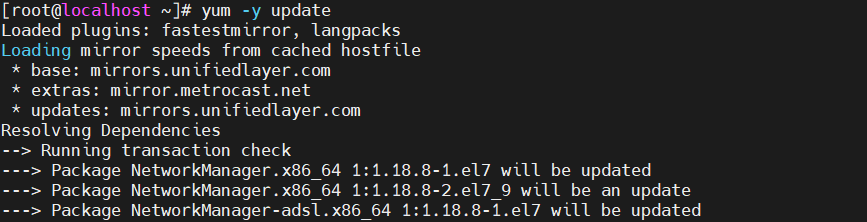


Fig. 1-Actualización de sistema

Se instala una serie de herramientas y utilidades comunes en el sistema, incluyendo Git, wget, curl, vim, bash-completion y tmux.

sudo yum -y install git wget curl vim bash-completion tmux

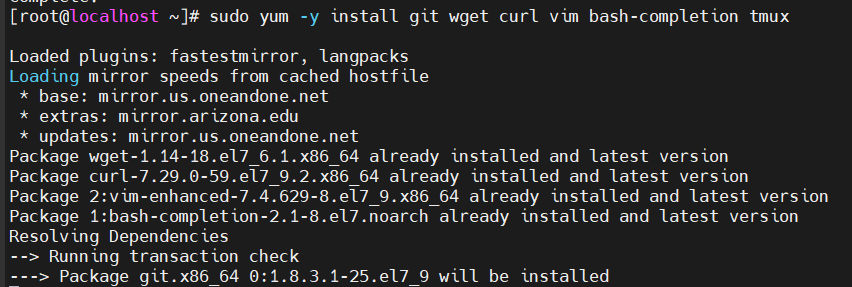


Fig. 2-Instalación de herramientas y utilidades

El siguiente comando proporciona paquetes adicionales para CentOS. Este repositorio es necesario para instalar Ansible, ya que no está disponible en los repositorios por defecto de CentOS.

sudo yum install epel-release

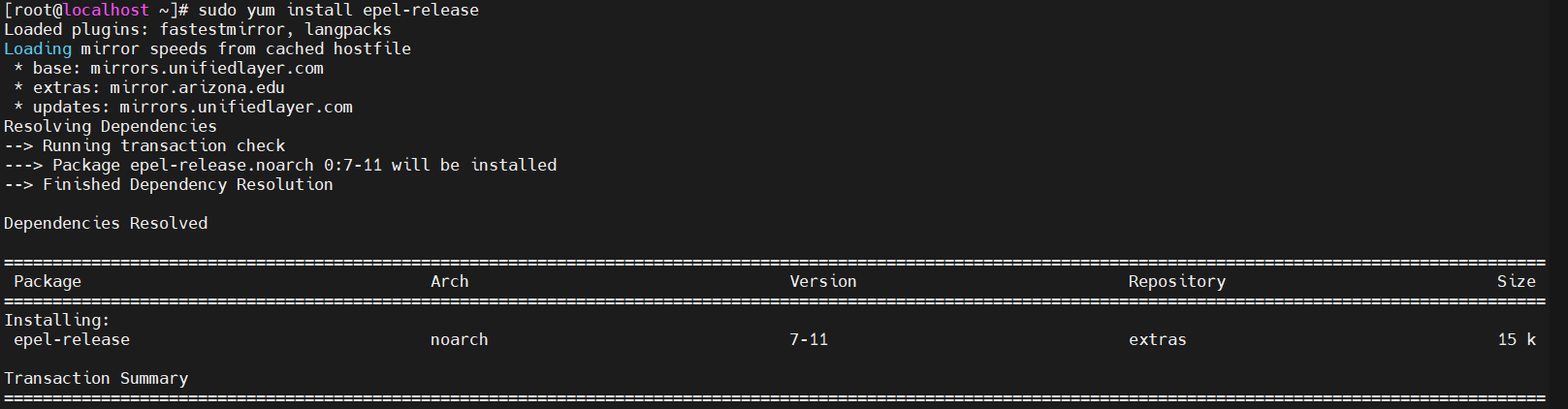


Fig. 3-Instalación de repositorio epel

Finalmente se instala Ansible, una herramienta de automatización de IT y gestión de configuraciones, en el sistema.

sudo yum install ansible

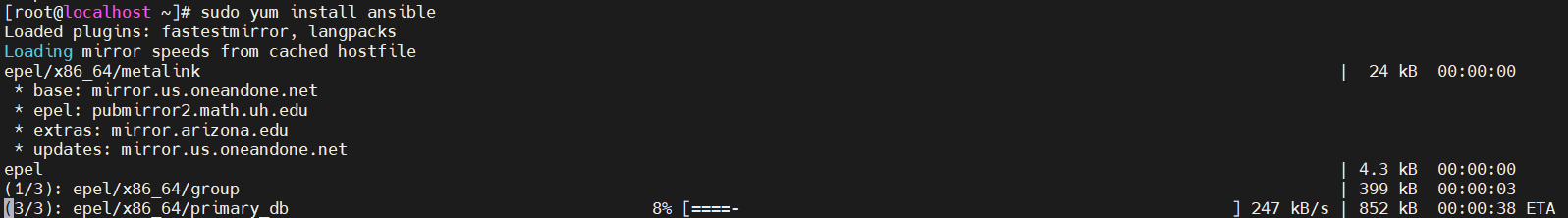


Fig. 4-Instalación ansible

Como último paso se comprueba la versión de Ansible instalada en el sistema. Este comando verifica que Ansible se haya instalado correctamente y muestra información sobre la configuración actual de Ansible, como la ubicación del archivo de configuración y la versión de Python utilizada.

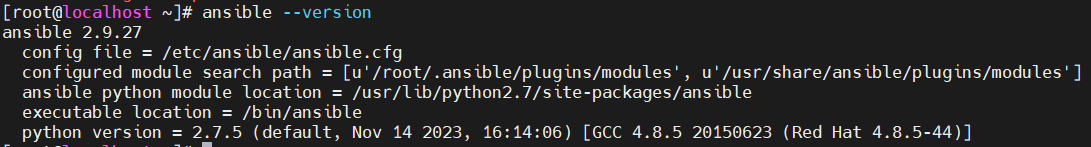


Fig. 5-Versión Ansible

Esto se realiza en el nodo master y los 2 nodos a configurar, la actualización como la instalación de ansible. Es importante terminar este proceso en las 3 máquinas.

* 1. **Configuración Ansible**

Primero se abre el archivo configuración de hosts de Ansible para agregar las direcciones IP de los servidores que se van a gestionar. En este caso, se añaden las direcciones IP de los nodos con un formato de lista. Y en el apartado de [dbservers] cambiar por [ALL] y posteriormente agregar las IP tanto del master como los nodos.

vi /etc/ansible/hosts



Fig. 6-Configuración de hosts

Se edita el archivo host del sistema para asociar las direcciones IP con los nombres de host de los servidores. Esto permite que los servidores se comuniquen entre sí utilizando nombres de host en lugar de direcciones IP.

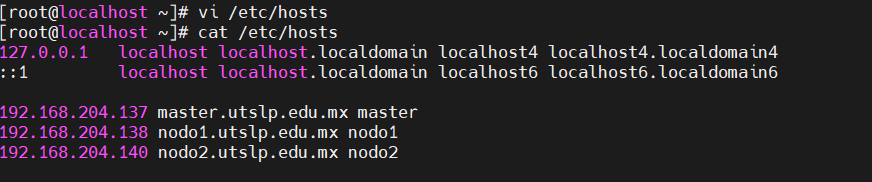


Fig. 7-Archivo de hosts del sistema

Para poder tener el control de las demás maquinas a las cuales el nodo master va mandar ordenes es necesario generar una llave ssh, la cual permite tener acceso a otros servidores. Con el comando:

ssh-keygen

*Enter file in which to save the key (/root/.ssh/id\_rsa):* Se pregunta en qué archivo se desea guardar la clave privada. Por defecto, se sugiere /root/.ssh/id\_rsa, pero se puede especificar otro archivo.

*Enter passphrase (empty for no passphrase):* En este paso, se solicita que se ingrese una contraseña para proteger la clave privada. Si se deja el campo vacío, la clave privada estará sin contraseña.

*Enter same passphrase again:* Para asegurarse de que no haya errores, se pide que se ingrese la misma contraseña de nuevo para confirmar.

*Your identification has been saved in /root/.ssh/id\_rsa:* Este mensaje confirma que la clave privada ha sido generada y guardada en el archivo especificado.

*Your public key has been saved in /root/.ssh/id\_rsa.pub*: Aquí se indica que la clave pública correspondiente a la clave privada generada ha sido guardada en el archivo /root/.ssh/id\_rsa.pub.

*The key fingerprint is:* Se muestra el fingerprint (huella digital) de la clave generada. Esta huella se utiliza para verificar la autenticidad de la clave pública cuando se establece una conexión SSH.

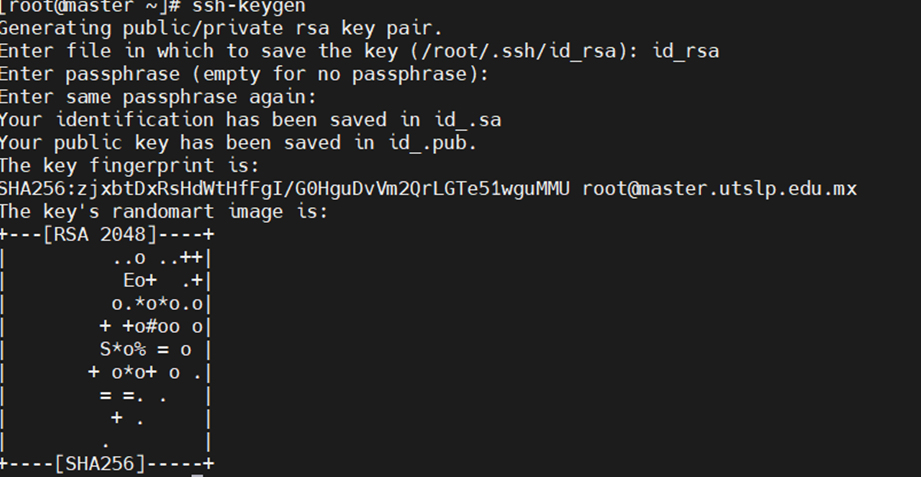


Fig. 8-Generar clave SSH

Finalmente, se copia la clave pública SSH al servidor remoto con la dirección IP 192.168.204.143 junto con las demás IP. Esto permite la autenticación sin contraseña al conectarse al servidor remoto como usuario root. Se repite este paso para cada servidor remoto que se va a gestionar.

ssh-copy-id root@192.168.204.143

ssh-copy-id root@192.168.204.142

ssh-copy-id root@192.168.204.138

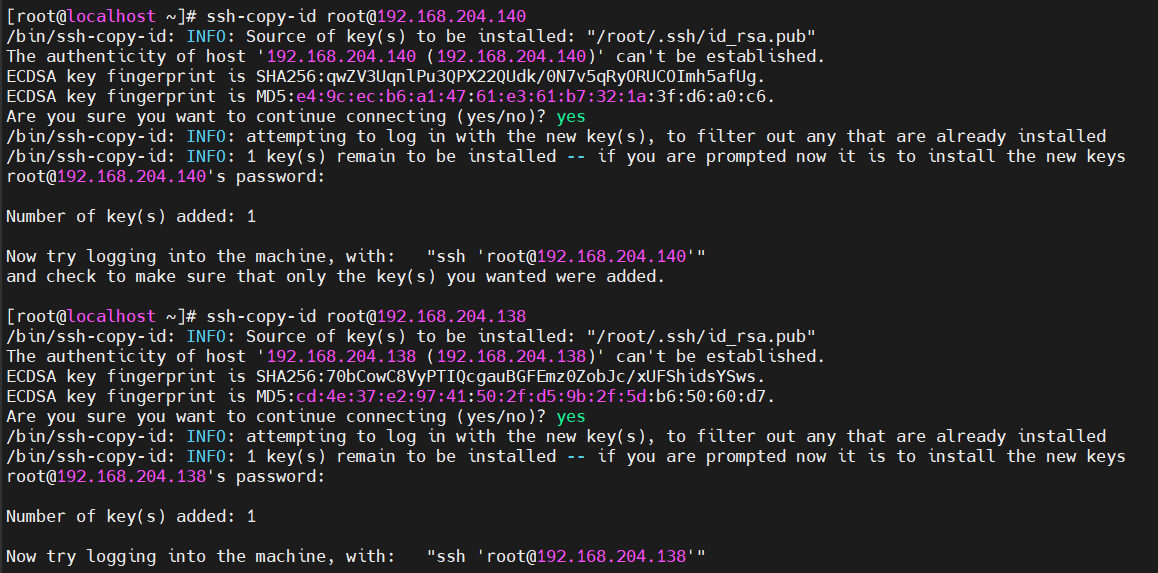


Fig. 9-Clave ssh de cada nodo agregada al master

También es de suma importancia cambiar el nombre de cada host individualmente con:

hostnamectl set-hostname master.utslp.edu.mx master

hostnamectl set-hostname nodo1.utslp.edu.mx nodo1

hostnamectl set-hostname nodo2.utslp.edu.mx nodo2

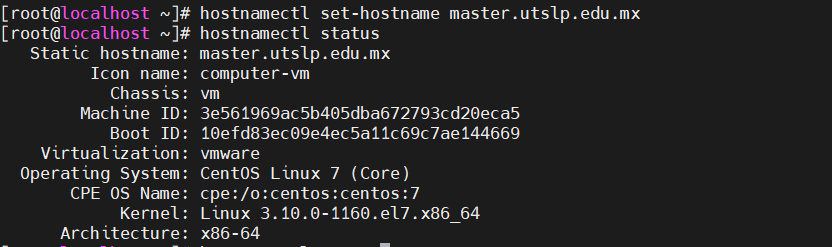


Fig. 10-Cambio de nombre de nodo master

Esto actualiza tanto el nombre de host estático como el nombre de host que se muestra en la terminal.

Cada comando que se ha realizado anteriormente es de suma importancia ingresarlo en los 3 nodos

* 1. **Instalación de Kubernetes (k8s)**

Se empieza clonando el repositorio de GitHub k8s-pre-bootstrap en el directorio actual. Este repositorio contiene scripts y configuraciones necesarias para la pre-configuración de un clúster de Kubernetes.

git clone https://github.com/jmutai/k8s-pre-bootstrap.git

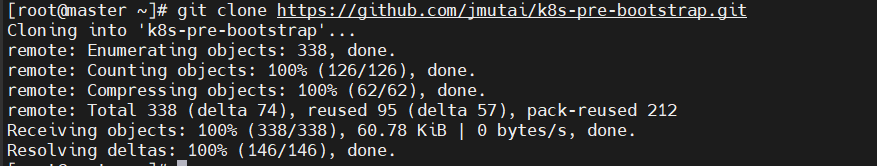


Fig. 11-Clonación de repositorio de Kubernetes

Se procede a cambiar al directorio k8s-pre-bootstrap que se acaba de clonar. Dentro de este directorio se encuentran los archivos y scripts necesarios para la preconfiguración.

cd k8s-pre-bootstrap

Por último, se abre el editor de texto vim para editar el archivo hosts. En este archivo se especifican las direcciones y los nombres de los nodos que formarán parte del clúster de Kubernetes. En este caso, se define los nodos del clúster.

[k8snodes] #Nombre de clúster

master #Master

nodo1 #Nodo1

nodo2 #Nodo2

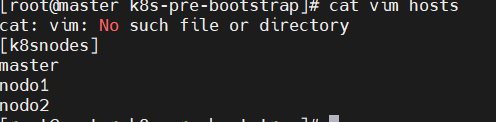


Fig. 12-Clústers definidos en hosts

* 1. **Playbook para instalar Kubernetes**

Este playbook de Ansible se utiliza para preparar los nodos de Kubernetes antes de la configuración del clúster. Define tareas y configuraciones que se aplicarán a los nodos especificados en el grupo k8snodes.

Como primer paso se crea y edita un archivo llamado k8s-prep.yml, que contendrá el playbook de Ansible para preparar los nodos de Kubernetes.

**Secciones del playbook:**

*name:* Prepare Kubernetes Nodes for Cluster bootstrapping - Nombre descriptivo de la tarea principal del playbook.

*hosts:* k8snodes - Especifica que el playbook se ejecutará en los nodos listados en el grupo k8snodes.

*remote\_user:* root - Usuario remoto que se utilizará para ejecutar las tareas.

*become:* yes - Especifica que se deben usar privilegios elevados (sudo) para las tareas.

*become\_method:* sudo - Método a utilizar para elevar privilegios.

**Variables:**

*k8s\_version:* "1.29" - Versión de Kubernetes que se instalará en los nodos.

*selinux\_state:* permissive - Estado de SELinux (permisivo) a configurar en los nodos.

*timezone:* "America/Mexico\_City" - Zona horaria que se establecerá en todos los nodos.

*k8s\_cni:* calico - Proveedor de red de contenedores (CNI), como Calico o Flannel.

*container\_runtime:* cri-o - Runtime de contenedores a utilizar (Docker, CRI-O, Containerd).

*pod\_network\_cidr:* "172.18.0.0/16" - Subred para los pods, si se utiliza CRI-O.

*configure\_firewalld:* false - Indica si se debe configurar firewalld (se recomienda false para versiones de k8s > 1.19).

*setup\_proxy:* false - Indica si se debe configurar un proxy para Docker.

*proxy\_server:* "proxy.example.com:8080" - Dirección y puerto del servidor proxy.

*docker\_proxy\_exclude:* "localhost,127.0.0.1" - Direcciones a excluir del proxy.

**Roles:**

*roles:* - kubernetes-bootstrap - Especifica el rol de Ansible kubernetes-bootstrap que se utilizará para ejecutar las tareas definidas en el playbook.

Al abrir se muestra el contenido archivo donde se va a cambiar la opción de timezone por “America/Mexico\_City” para después guardar y se realice el cambio.

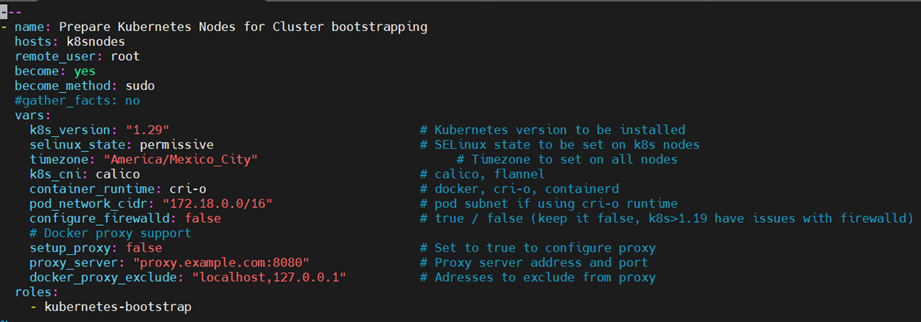


Fig. 13-Contenido del archivo 'k8s-prep-yml'

* 1. **Revisión del servicio SSH**

Se edita el archivo de configuración de SSH con:

vi /etc/ssh/sshd\_config

Dentro del archivo se agrega. Este ajuste permite que el usuario root inicie sesión directamente a través de SSH.

PermitRootLogin yes

Se reinicia el servicio SSH para aplicar los cambios realizados en el archivo de configuración.

systemctl restart ssh

Por último, se establece o cambia la contraseña del usuario root. Al ejecutarlo, se te solicita que se ingrese y confirme la nueva contraseña para el usuario root.

passwd root

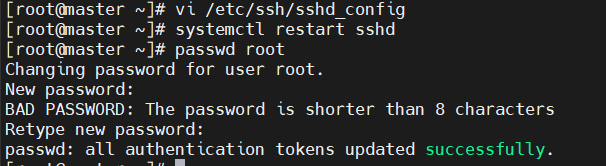


Fig. 14-Comandos aplicados a nodo master

* 1. **Ejecución de Playbook**

Este proceso asegura que los servidores están accesibles y que el Playbook de Ansible está libre de errores de sintaxis antes de proceder a su ejecución, lo cual es fundamental para una implementación exitosa y sin problemas del clúster de Kubernetes.

Como primer paso se ejecuta la conexión por SHH hacia los servidores utilizando el nombre del mismo. Esto se realiza en

ssh master

ssh nodo1

ssh nodo2

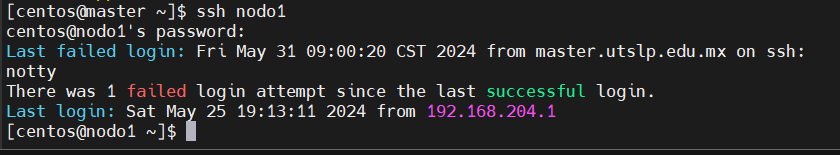


Fig. 15-Conexión SHH de master hacia nodo1

Se procede a verificar la sintaxis del archivo Playbook k8s-prep.yml sin ejecutar realmente ninguna tarea. Utiliza el inventario de hosts definido en el archivo hosts. Primero se ingresa al directorio actual de trabajo y a continuación la confirmación de sintaxis.

cd k8s-pre-bootstrap/

ansible-playbook --syntax-check -i hosts k8s-prep.yml

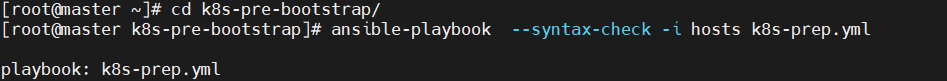


Fig. 16-Verificando si es correcta la sintaxis

Por consiguiente, se inicia el agente SSH y se agrega la clave SSH usando:

eval `ssh-agent -s` && ssh-add

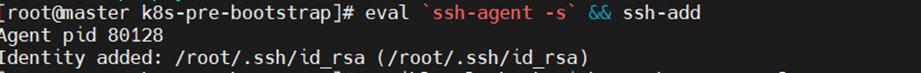


Fig. 17-Inicio de agente y clave SSH

El agente SSH maneja tus claves SSH y facilita la autenticación a múltiples servidores sin tener que ingresar la contraseña cada vez. Esto es útil para ejecutar comandos Ansible que requieren autenticación SSH en varios nodos.

Finalmente se ejecuta el playbook de Ansible k8s-prep.yml utilizando el inventario especificado en el archivo hosts. Este playbook configura los nodos para el clúster de Kubernetes según las especificaciones definidas en k8s-prep.yml.

ansible-playbook -i hosts k8s-prep.yml

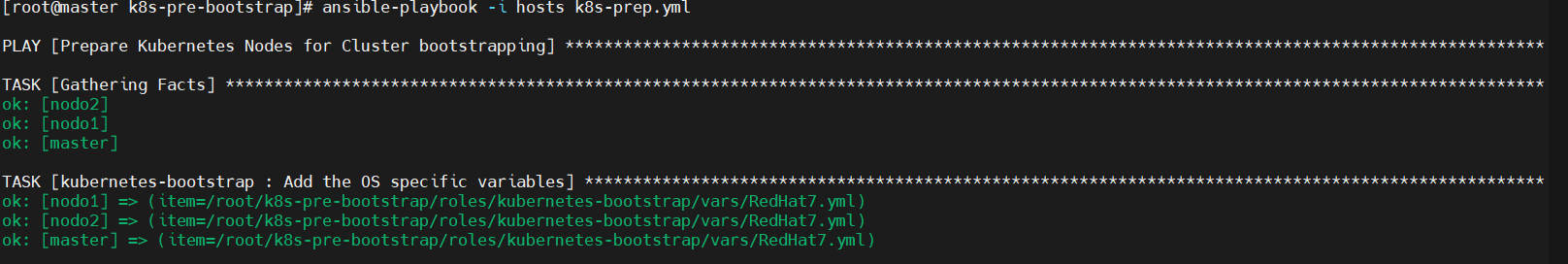


Fig. 18-Ejecución de Playbook

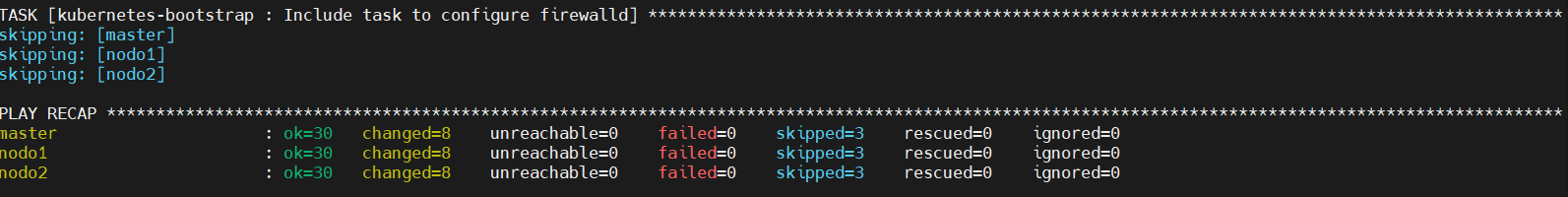


Fig. 19-Ejecución exitosa de los 3 nodos

* + 1. **Errores al ejecutar Playbook de Ansible**

*Error módulo ansible.posix.selinux*

Al realizar ejecución del comando anterior resultaron varios errores que vienen con su descripción y solución en el siguiente apartado:

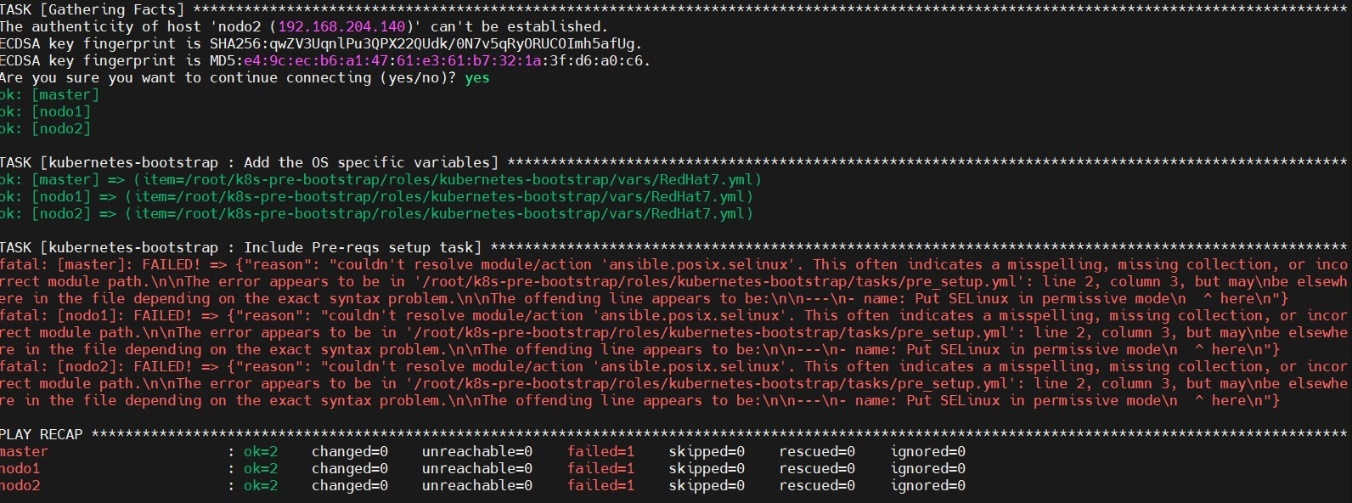


Fig. 20-Error de módulo ansible.posix.selinux

El error indica que Ansible no puede encontrar el módulo ansible.posix.selinux. Esto generalmente ocurre por una de las siguientes razones:

* El módulo no está instalado.
* El nombre del módulo está mal escrito.
* La colección que contiene el módulo no está instalada.

Para su solución se necesita entrar a la carpeta

cd roles/kubernetes-bootstrap/tasks/

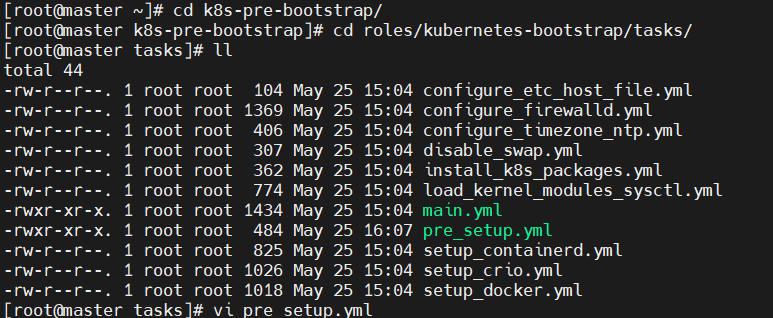


Fig. 21-Búsqueda en directorio de archivo pre\_setup.yml

Para después editar el archivo pre\_setup.yml para comentar la tarea de selinux, las primeras 4 líneas omitiendo el signo --- se guarda el archivo y se ejecuta nuevamente el playbook.

vi pre\_setup.yml

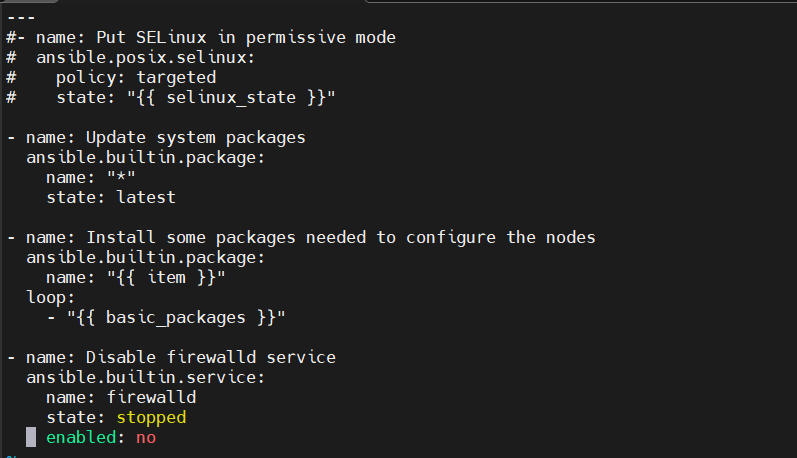


Fig. 22-Archivo comentado en las 4 primeras líneas

*Error de módulo timezone*

Sugiere que Ansible no puede encontrar el módulo `timezone` de la colección `community.general`. Para resolver este problema, se debe instalar la colección `community.general`.

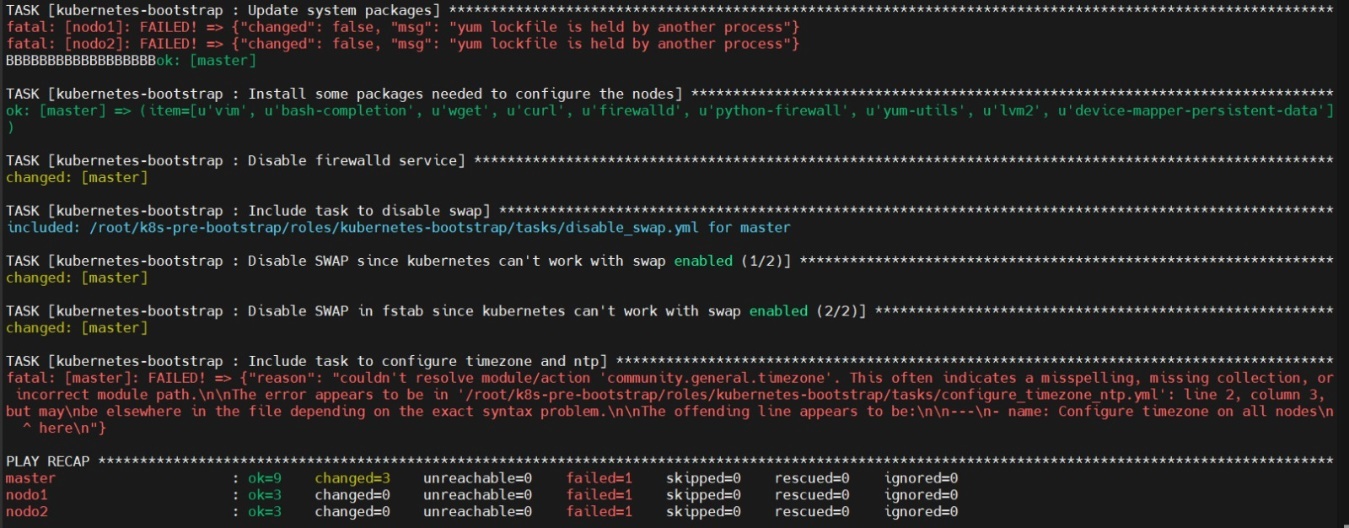


Fig. 23-Error de community.general

Por lo que se tiene que instalar:

ansible-galaxy collection install community.general

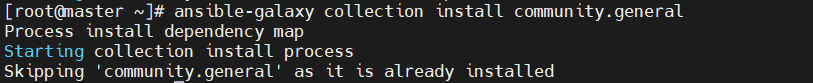


Fig. 24-Instalación de módulo community.general

1. **REVISIÓN DE SERVICIOS**

Para verificar el estado y la configuración de los servicios necesarios después de ejecutar el playbook de Ansible y ya tener correctamente el funcionamiento.

1. Primero se verifica el estado de servicio CRI-O, el cual debe estar activo.

systemctl status crio

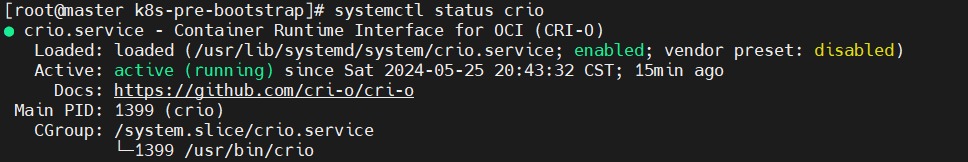


Fig. 25-Servicio crio activo correctamente

1. Se verifica el estado del firewall que tiene que estar desactivado.

systemctl status firewalld

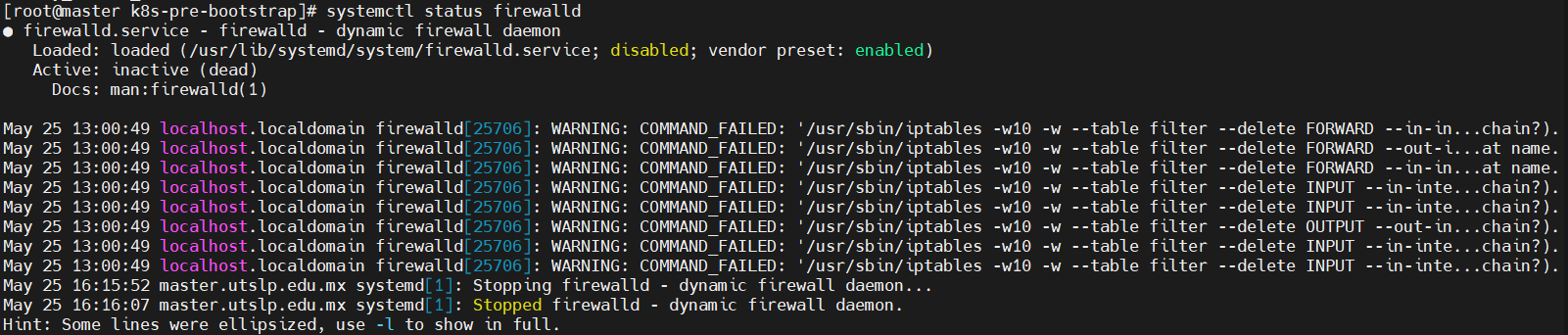


Fig. 26-Firewall desactivado

Estos comandos te permiten verificar que los servicios cruciales (CRI-O y firewall) y las configuraciones del sistema están correctamente aplicados y en funcionamiento. Esto es esencial para asegurarte de que el entorno esté preparado para la instalación y configuración de Kubernetes.