

# Documentación sobre la Instalación y Configuración de Ray para Ejecución Distribuida

## 1. Configuración de las Máquinas Virtuales

- **Máquinas utilizadas:** Se implementaron tres máquinas virtuales: Windows 10 (64 bits), Ubuntu y Linux Mint.
- **Redes:** Cada máquina se configuró en modo "solo anfitrión" en VirtualBox para garantizar la comunicación entre ellas sin necesidad de una conexión a internet externa.

## 2. Instalación de Python y Pip

- Se verificó la instalación de Python (utilizando `python --version`) y, si era necesario, se instaló la versión adecuada.
- En Linux, se instalaron paquetes esenciales con:

```
sudo apt update
sudo apt install python3 python3-pip -y
```

## 3. Instalación de Ray

- Se creó un entorno virtual en cada máquina para evitar conflictos:

```
python3 -m venv ray-env
source ray-env/bin/activate # Linux/Mac
.\ray-env\Scripts\activate # Windows
```

- ❖ Ray se instaló dentro del entorno virtual:

```
pip install ray
```

## 4. Iniciación del Clúster de Ray

- En la máquina principal (Linux Mint), se ejecutó:

```
ray start --head --port=6379
```

- En las máquinas secundarias (Ubuntu), se conectaron al nodo principal:

```
ray start --address='IP_DEL_NODO_PRINCIPAL:6379'
```

## 5. Verificación de Conexión

- Se usó el comando `ray status` en el nodo principal para verificar si los nodos secundarios se habían unido al clúster.

## 6. Ejecución del Script de Prueba

- El script Python se ejecutó en el nodo principal y distribuyó tareas en los nodos secundarios:

```
import ray
ray.init(address='auto') # Se conecta al nodo principal automáticamente

@ray.remote
def tarea_distribuida(x):
    return x * x

resultados = ray.get([tarea_distribuida.remote(i) for i in range(4)])
print("Resultados:", resultados)
```

**Resultados Esperados:** El script imprimirá el nombre único del nodo donde se ejecutó cada tarea, lo que indica que las tareas se distribuyeron correctamente entre las máquinas.

## Solución de Problemas Comunes:

- **Problemas de red:** Verificar la conectividad entre máquinas con `ping`.
- **Conexión de nodos:** Asegurarse de que los puertos estén abiertos y accesibles.

- **Errores en Python:** Usar entornos virtuales y versiones compatibles de Python y Ray.