**Hyperledger Indy Production Setup Guide**

This guide will walk you through setting up a production-level Hyperledger Indy network with 4 nodes and 5 clients. You'll also learn how to create DIDs for an Aries Agent. Each step includes a detailed explanation to help you understand the purpose and expected outcome.

**Setup Configuration**

**Install System-Wide Dependencies**

sudo apt update

sudo apt install unzip python3-venv python3-pip -y

**Create and Activate a Virtual Environment for Both Indy Plenum & Indy Node**

mkdir indy-setup && cd indy-setup

python3 -m venv venv-indy

source venv-indy/bin/activate

**Install Latest Indy and Plenum**

1. **Add the Indy Repository:**

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 9692C00E657DDE61

sudo add-apt-repository "deb https://hyperledger.jfrog.io/artifactory/indy focal stable"

**Explanation:** This adds the Hyperledger Indy repository to your system, allowing you to download and install the latest versions of Indy and its dependencies.

**Install Plenum Dependencies**

1. **Download and Install Dependencies:**

wget http://security.ubuntu.com/ubuntu/pool/main/o/openssl1.0/libssl1.0.0\_1.0.2n-1ubuntu5.13\_amd64.deb

wget https://repo.sovrin.org/deb/pool/bionic/master/u/ursa/ursa\_0.3.2-1\_amd64.deb

sudo dpkg -i libssl1.0.0\_1.0.2n-1ubuntu5.13\_amd64.deb

sudo dpkg -i ursa\_0.3.2-1\_amd64.deb

**Explanation:** These commands download and install essential dependencies like libssl and ursa, which are required for cryptographic operations in Indy.

**Install Indy Plenum**

1. **Setup Indy Plenum Directory:**

mkdir indy-plenum && cd indy-plenum

**Explanation:** This creates a directory to store Indy Plenum files and navigates into it.

1. **Download and Install Plenum:**

wget https://github.com/hyperledger/indy-plenum/releases/download/v1.13.1/third-party-dependencies.zip

wget https://github.com/hyperledger/indy-plenum/releases/download/v1.13.1/plenum-deb.zip

sudo apt install unzip

unzip third-party-dependencies.zip -d third-party-dependencies

cd third-party-dependencies/artifacts/third-party-dependencies && sudo dpkg -i \*.deb

(Note:->if facing error like this then dpkg: error processing package rocksdb (--install):

dependency problems - leaving unconfigured

Setting up python3-dateutil (2.8.2) ...

Errors were encountered while processing:

python3-packaging

python3-prompt-toolkit

python3-ursa

rocksdb

then:

#

apt --fix-broken install

)

cd ../../../

sudo unzip plenum-deb.zip -d plenum-deb

cd plenum-deb/artifacts/plenum-deb

sudo dpkg -i \*.deb

cd ../../../..

sudo mv /usr/lib/ursa/\* /usr/lib

sudo rm -rf /usr/lib/ursa

**Explanation:** This step downloads and installs Indy Plenum, which is responsible for transaction processing and consensus. It also sets up third-party dependencies required by Plenum.

**Install Indy Node**

1. **Setup Indy Node Directory:**

mkdir indy-node && cd indy-node

**Explanation:** This creates a directory to store Indy Node files and navigates into it.

1. **Download and Install Node Dependencies:**

wget https://github.com/hyperledger/indy-node/releases/download/v1.13.2/third-party-dependencies.zip

unzip third-party-dependencies.zip -d third-party-dependencies

cd third-party-dependencies/artifacts/third-party-dependencies

sudo dpkg -i \*.deb

cd ../../../

**Explanation:** This step installs additional third-party dependencies required by Indy Node.

1. **Install Supervisor:**

sudo apt-get install supervisor

**Explanation:** Supervisor is installed to manage and monitor the Indy Node processes.

1. **Download and Install Indy Node:**

wget https://github.com/hyperledger/indy-node/releases/download/v1.13.2/indy\_node-deb.zip

unzip indy\_node-deb.zip -d indy\_node

cd indy\_node/artifacts/indy\_node-deb

sudo dpkg -i \*.deb

**Explanation:** This installs Indy Node, which forms the core of the network by handling transactions, consensus, and communication between nodes.

**Creating a Network of 4 Nodes**

1. **Network name setup:**

sudo nano /etc/indy/indy\_config.py

NETWORK\_NAME = 'sandbox'

Or any desired name

**Generate Pool Transactions:**

sudo generate\_indy\_pool\_transactions --nodes 4 --clients 5 --nodeNum 1 2 3 4

**Explanation:** This command generates the necessary transactions to initialize a pool of 4 nodes and 5 clients. It creates the configuration files required for the network to function.

**Installing Indy CLI for Agent Registration**

1. **Download and Install Indy CLI:**

**Create a IndyCli directory in the same directory as indy-node and indy plenum**

mkdir indy-cli

cd indy-cli

wget https://github.com/hyperledger/indy-cli-rs/releases/download/v0.1.0/indy-cli-rs-0.1.0-linux-x86\_64.tar.gz

tar -xf indy-cli-rs-0.1.0-linux-x86\_64.tar.gz

chmod +x indy-cli-rs

./indy-cli-rs

**Explanation:** Indy CLI is a command-line tool used to interact with the Indy network, register agents, create wallets, and manage DIDs.

1. **Create and Open Wallet:**

wallet create <wallet\_name> key=<wallet\_password>

wallet open <wallet\_name> key=<wallet\_password>

ex->

wallet create trustee key=trustee

wallet open trustee key=trustee

**Explanation:** This creates a secure wallet for storing cryptographic keys and DIDs, then opens it for use.

1. **Generate DID:**

did new seed=<32\_digit\_hex\_string>

ex-> did new seed=000000000000000000000000Trustee1 (trustee seed )

wallet close

exit

**Explanation:** This generates a new DID (Decentralized Identifier) using a 32-digit hexadecimal seed, providing a unique identity on the network.

**Modify Genesis Pool File**

(Optional : if you want to insert your own did as trustee by default the trustee seed is 000000000000000000000000Trustee1 )

1. **Edit Domain Genesis Pool File:**

sudo nano /var/lib/indy/sandbox/domain\_transactions\_genesis

**Explanation:** Modify the genesis file to include your new DID and verkey. This ensures your identity is recognized and trusted by the network.

**Starting Nodes**

1. **Start Nodes in Background: (By default the port start from 9701 )**

sudo start\_indy\_node Node1 0.0.0.0 9701 0.0.0.0 9702 &

sudo start\_indy\_node Node2 0.0.0.0 9703 0.0.0.0 9704 &

sudo start\_indy\_node Node3 0.0.0.0 9705 0.0.0.0 9706 &

sudo start\_indy\_node Node4 0.0.0.0 9707 0.0.0.0 9708 &

**Explanation:** These commands start the 4 Indy nodes in the background, allowing them to begin processing transactions and communicating with each other.

1. **Check Node Logs:**

cd /var/log/indy/sandbox

tail -f <Node>.log

**Explanation:** Checking the log files helps you verify that the nodes are running correctly and identify any issues.

**Registering Agent DID to Ledger**

1. **Run Indy CLI:**

**cd <indy cli directory>**

./indy-cli-rs

**Explanation:** Launches the Indy CLI for interacting with the network.

1. **Create Pool to Connect Ledger:**

pool create <pool\_name> gen\_txn\_file=/var/lib/indy/sandbox/pool\_transactions\_genesis

pool connect <pool\_name>

**Explanation:** This creates and connects to a pool, allowing you to interact with the ledger.

1. **Open Trustee Wallet:**

wallet open <wallet\_name> key=<wallet\_key>

wallet open trustee key=trustee

**Explanation:** Opens the trustee wallet, which has the necessary permissions to register new DIDs and make changes to the ledger.

1. **Use Trustee DID:**
2. By Default the ledger seed is 000000000000000000000000Trustee1 (if not modified in step 14)

did new seed=000000000000000000000000Trustee1

did use <trustee\_did>

**Explanation:** This sets the active DID to the trustee DID, giving you the necessary authority to make ledger changes.

1. **Create New Agent DID:**

did new seed=<32\_hex\_string>

**Explanation:** Generates a new DID for an agent, which will be used to interact with the ledger and other agents.

1. **Send DID to Ledger:**

ledger nym did=<new\_agent\_did> verkey=<new\_agent\_verkey> role=<desired\_role>

**Explanation:** Registers the new agent DID on the ledger with the specified role, enabling it to participate in the network.

**Agent Role code in Indy**

**-------------------------**

**0 - TRUSTEE**

**2 - STEWARD**

**101 - TRUST\_ANCHOR**

**101 - ENDORSER - equal to TRUST\_ANCHOR that will be removed soon**

**201 - NETWORK\_MONITOR**