You can't directly create a true 3-node Kubernetes cluster on a single local machine *without virtualization*. Kubernetes requires multiple independent machines (physical or virtual) to function as a cluster. However, you *can* simulate a multi-node setup for learning and development using tools like Minikube or kind (Kubernetes IN Docker).

Here's how to achieve a similar experience using Minikube, which is the most common and recommended approach for local Kubernetes development:

**Using Minikube (Simulating a Multi-Node Experience):**

Minikube creates a single-node cluster inside a virtual machine. While it's technically one node, it allows you to deploy applications and services as if you were on a multi-node cluster, demonstrating the core Kubernetes concepts.

1. **Prerequisites (Local Machine Setup):**
   * **Operating System:** Windows, macOS, or Linux.
   * **Virtualization:** A hypervisor (VirtualBox, Hyper-V, VMware, or Docker driver) is required. Docker Desktop includes Docker as a driver.
   * **kubectl:** The Kubernetes command-line tool.
   * **Minikube:** The Minikube tool.
2. **Installation:**

Follow the installation instructions for your operating system on the official Minikube website: [https://minikube.sigs.k8s.io/docs/start/](https://www.google.com/url?sa=E&source=gmail&q=https://minikube.sigs.k8s.io/docs/start/)

1. **Start Minikube:**

Bash

minikube start

This command starts a single-node Kubernetes cluster inside a VM. If you have multiple hypervisors installed, you might need to specify the driver:

Bash

minikube start --driver=<driver\_name> # e.g., minikube start --driver=virtualbox

1. **Create an NGINX Deployment (3 Replicas):**

Create a file named nginx-deployment.yaml:

YAML

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

1. **Apply the Deployment:**

Bash

kubectl apply -f nginx-deployment.yaml

1. **Create a NodePort Service:**

Create a file named nginx-service.yaml:

YAML

apiVersion: v1

kind: Service

metadata:

name: nginx-service

spec:

type: NodePort

selector:

matchLabels:

app: nginx

ports:

- protocol: TCP

port: 80

targetPort: 80

nodePort: 30080 # Choose a port in the 30000-32767 range (or let Kubernetes auto-assign)

1. **Apply the Service:**

Bash

kubectl apply -f nginx-service.yaml

1. **Verify the Deployment and Service:**

Bash

kubectl get deployments

kubectl get pods

kubectl get services

1. **Access the NGINX Service:**
   * **Get the Minikube IP:**

Bash

minikube ip

* + **Access in Browser:** Open your web browser and go to http://<minikube\_ip>:<nodePort> (e.g., http://192.168.49.2:30080). You should see the NGINX welcome page.

**Key Points and Clarifications:**

* **Simulated Multi-Node:** Minikube simulates a multi-node environment on a single machine. It doesn't provide true multi-node functionality in the sense of distributing workloads across separate physical or virtual machines.
* **NodePort:** The NodePort service type exposes the service on a static port on each node's IP. In Minikube's case, it's the Minikube VM's IP.
* **Port Range:** NodePort uses ports in the range 30000-32767 by default.
* **Production vs. Local Development:** NodePort is generally not recommended for production environments. Use LoadBalancer (if your cloud provider supports it) or Ingress controllers for production.
* **Alternative: kind:** If you want a more "true" multi-node experience locally (using Docker containers as nodes), you can explore kind (Kubernetes IN Docker). However, Minikube is generally simpler for getting started.

This approach provides a practical way to learn about Kubernetes deployments and services on your local machine. If you need a real multi-node cluster, you'll need to use cloud providers or set up virtual machines.