Steps to Set Up a VM for Blockchain Projects.

🧩 Step 1: Update your system

```
sudo apt update && sudo apt upgrade -y
```

🧩 Step 2: Install curl (if not already installed)

```
sudo apt install curl -y
```

🧩 Step 3: Download and install NVM

Run this command (this fetches the latest version directly from GitHub):

```
curl -o-
https://raw.githubusercontent.com/nvm-sh/nvm/master/install.sh | bash
```

> ✅ This script clones the NVM repo to `~/.nvm` and updates your shell profile.

🧩 Step 4: Load NVM into your current terminal session

After installation, run:

```
export NVM_DIR="$([ -z "${XDG_CONFIG_HOME-}" ] && printf %s
"${HOME}/.nvm" || printf %s "${XDG_CONFIG_HOME}/nvm")"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
```

> Alternatively, **close and reopen your terminal** — the install script adds the above lines to your `~/.bashrc` or `~/.zshrc` automatically.

🧩 Step 5: Verify installation

```
nvm --version
```

If you see a version number (e.g., `0.39.7`), NVM is installed successfully ✅.

🧩 Step 6: Install a Node.js version

Example:

```
nvm install --lts     # Install latest LTS version
nvm install 20        # Install Node.js v20
```

🧩 Step 7: Use a specific version

```
nvm use 20
```

Make it the default for new terminals:

```
nvm alias default 20
```

🧩 Optional: Check all installed Node versions

```
nvm ls
```

🐳 Step-by-Step Installation Guide for Docker + Docker Compose

---

🧩 Step 1: Update system

```
sudo apt update && sudo apt upgrade -y
```

🧩 Step 2: Uninstall old Docker versions (optional but recommended)

```
sudo apt remove docker docker-engine docker.io containerd runc -y
```

🧩 Step 3: Install required dependencies

```
sudo apt install ca-certificates curl gnupg lsb-release -y
```

🧩 Step 4: Add Docker's official GPG key

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
--dearmor -o /etc/apt/keyrings/docker.gpg
```

🧩 Step 5: Add Docker repository

```
echo \
  "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.gpg] \
  https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

🧩 Step 6: Install Docker Engine and CLI

```
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin -y
```

> ✅ This installs both **Docker Engine** and **Docker Compose v2** (the plugin version).

🧩 Step 7: Verify Docker installation

```
sudo docker --version
```

```
sudo docker run hello-world
```

You should see a “Hello from Docker!” message.

🧩 Step 8: Manage Docker as a non-root user (optional but recommended)

```
sudo usermod -aG docker $USER
```

Then log out and log back in, or run:

```
newgrp docker
```

Now test:

```
docker ps
```

If it works **without sudo**, your setup is correct ✅

🧩 Step 9: Verify Docker Compose

Docker Compose is included as:

```
docker compose version
```

Example output:

```
Docker Compose version v2.27.0
```

🧩 (Optional) Step 10: Enable Docker on startup

```
sudo systemctl enable docker
sudo systemctl start docker
```

💡 Bonus Tip:

If you prefer the **standalone Compose v1 binary**, you can install it like this:

```
sudo curl -L
"https://github.com/docker/compose/releases/latest/download/docker-com
pose-$(uname -s)-$(uname -m)" \
-o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose
docker-compose --version
```

But the **plugin version (Step 6)** is recommended for newer systems.