

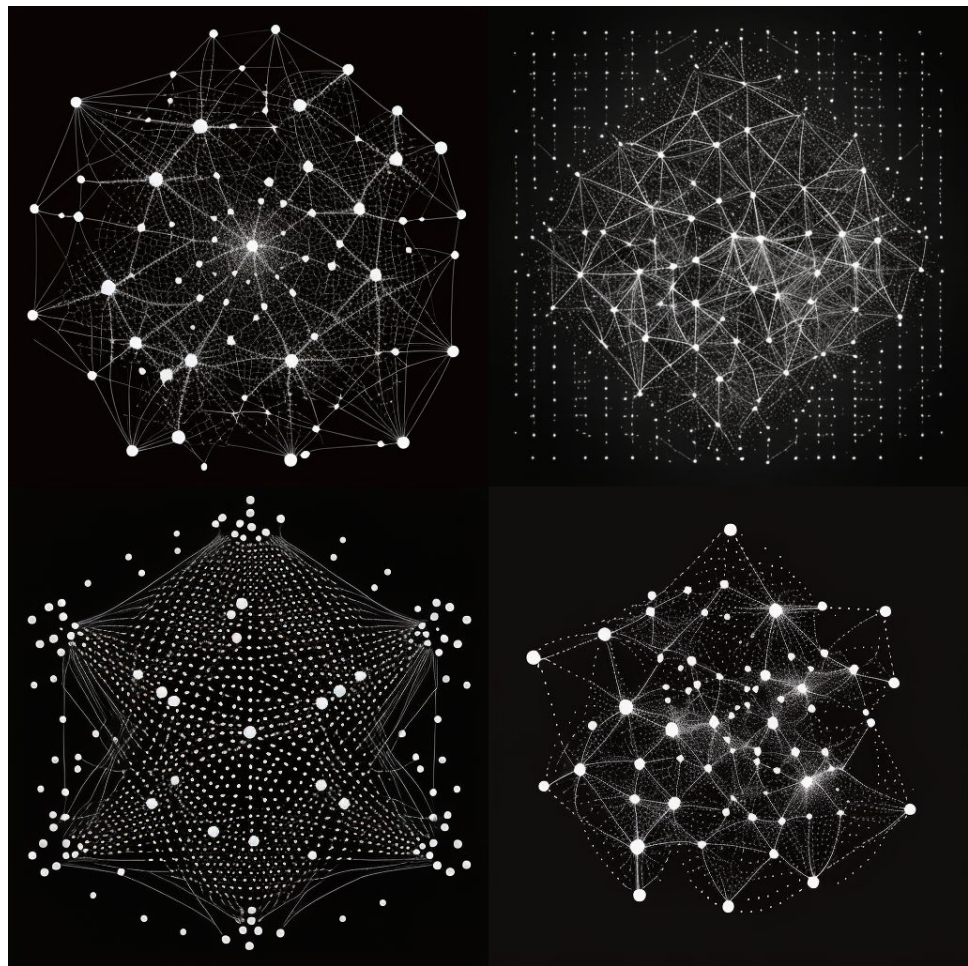
# Lesson 6: Intro to the Lightning Network



Theory / Implementations / protocol vs app dev / Lightning  
Development environments

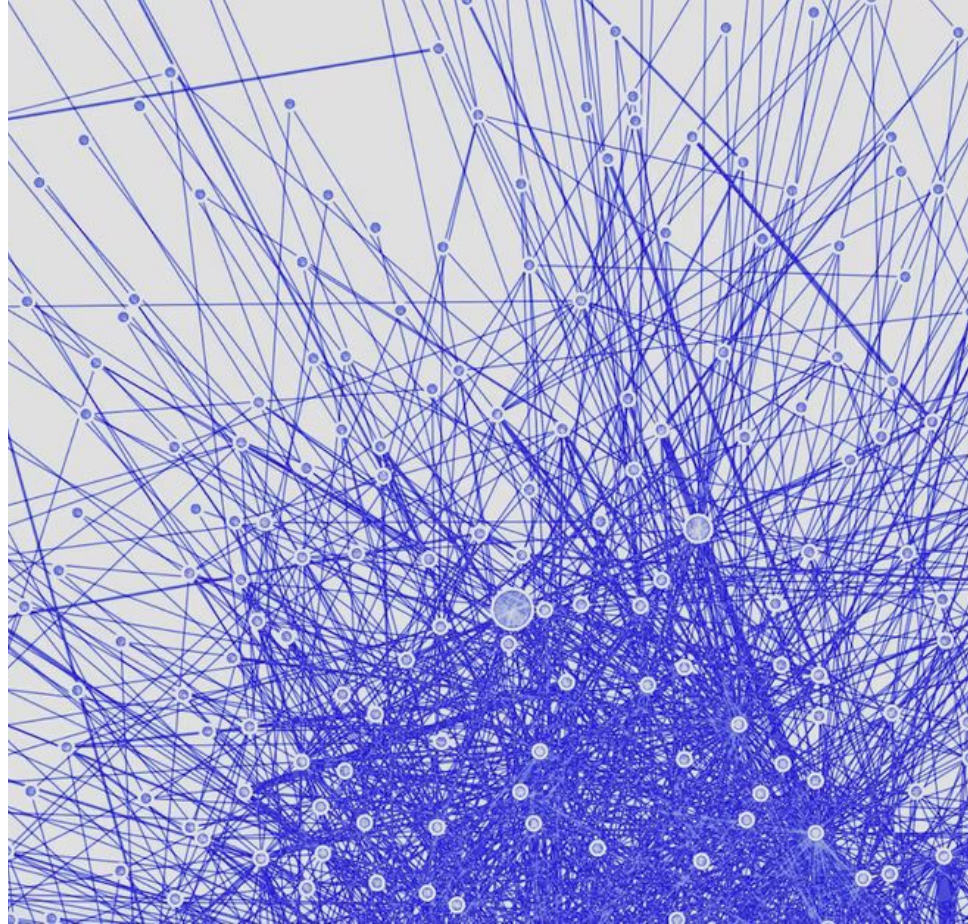
# The Lighting Network

- **Second-layer solution:** Built on top of Bitcoin's blockchain for faster, more scalable transactions
- **Off-chain transactions:** Enables payments between users without recording every transaction on the blockchain
- **Micropayments:** Allows for small, instant transactions with very minimal fees, expanding the range of possible use cases for Bitcoin



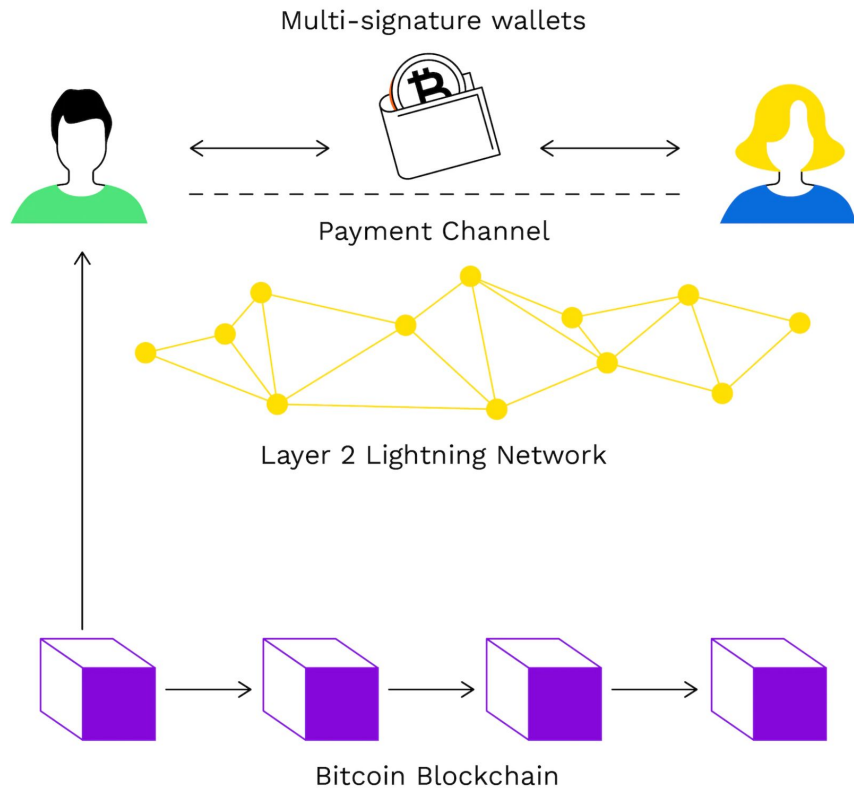
# Lightning Nodes

- **Network participants:** Lightning nodes are computers that participate in the Lightning Network by running compatible software
- **Routing payments:** Nodes help route transactions through the network by forwarding payments between channels
- **Decentralization:** A large number of nodes ensures the network remains decentralized and resistant to censorship or control by a single entity



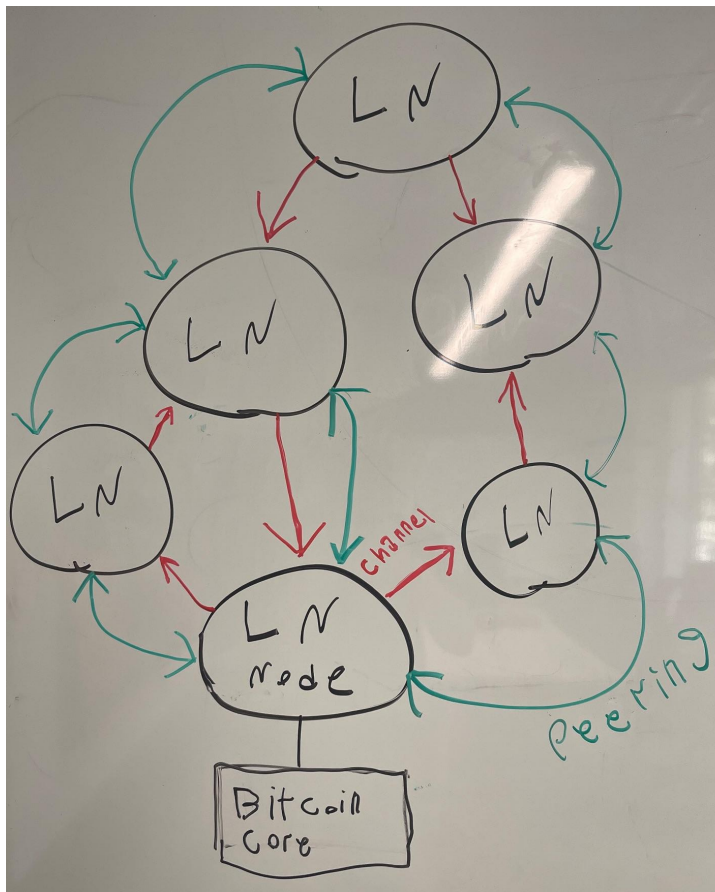
# Lightning Channels

- **Payment channels:** Temporary, private channels between users allow for multiple transactions without requiring on-chain confirmations
- **Multi-signature wallets:** Both parties in a channel have control over funds, ensuring security and trust
- **Network of channels:** Users can route payments through multiple channels, even if they don't have a direct channel with the recipient



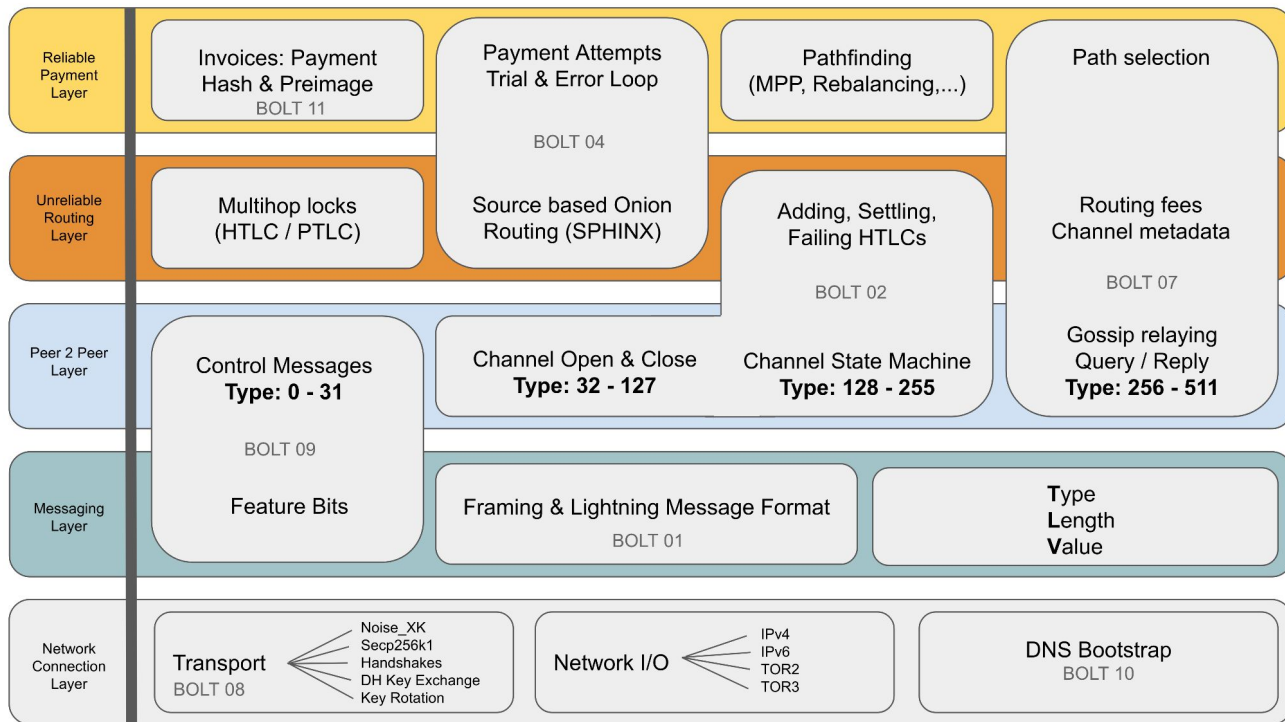


# Lets rip it on the whiteboard



# What is a Lightning implementation

A Lightning Network implementation is a software package that implements the Lightning spec, allows nodes to participate in the Lightning Network, and provides the necessary functionalities for a Lightning node



# Protocol dev VS App dev

PlebDevs course #1  
Frontend

User Interfaces (Mobile, Browser)



PlebDevs course #2  
Backend

Application Server



Protocol Developer  
Lightning

Lightning Network Node



Protocol Developer  
Bitcoin

Bitcoin Node



# The different Lightning Implementations

## - LND:

- Stands for “Lightning Network Daemon”
- The most popular / widely used LN implementation
- Developed by Lightning Labs, written in Go
- Rich feature set and extensive documentation



## - CoreLightning:

- Developed by Blockstream
- Written in C, optimized for performance and reliability
- Flexible plugin system for extended functionality
- Feature rich



## - Eclair:

- Developed by ACINQ
- Written in Scala, Highly scalable
- User-friendly wallet app and mobile SDK for app development
- Offers additional features such as channel management tools



## - LDK:

- Created by Spiral (formally Square crypto)
- Written in Rust, focusing on safety and performance
- Modular, customizable toolkit for building Lightning implementations and applications
- Enables seamless integration with various Bitcoin wallets and applications

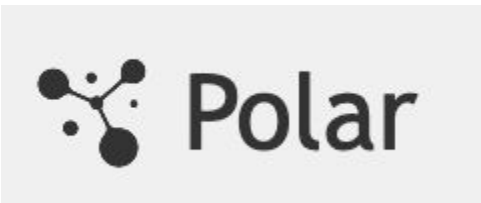




# Setting up a local Lightning Development Environment

---

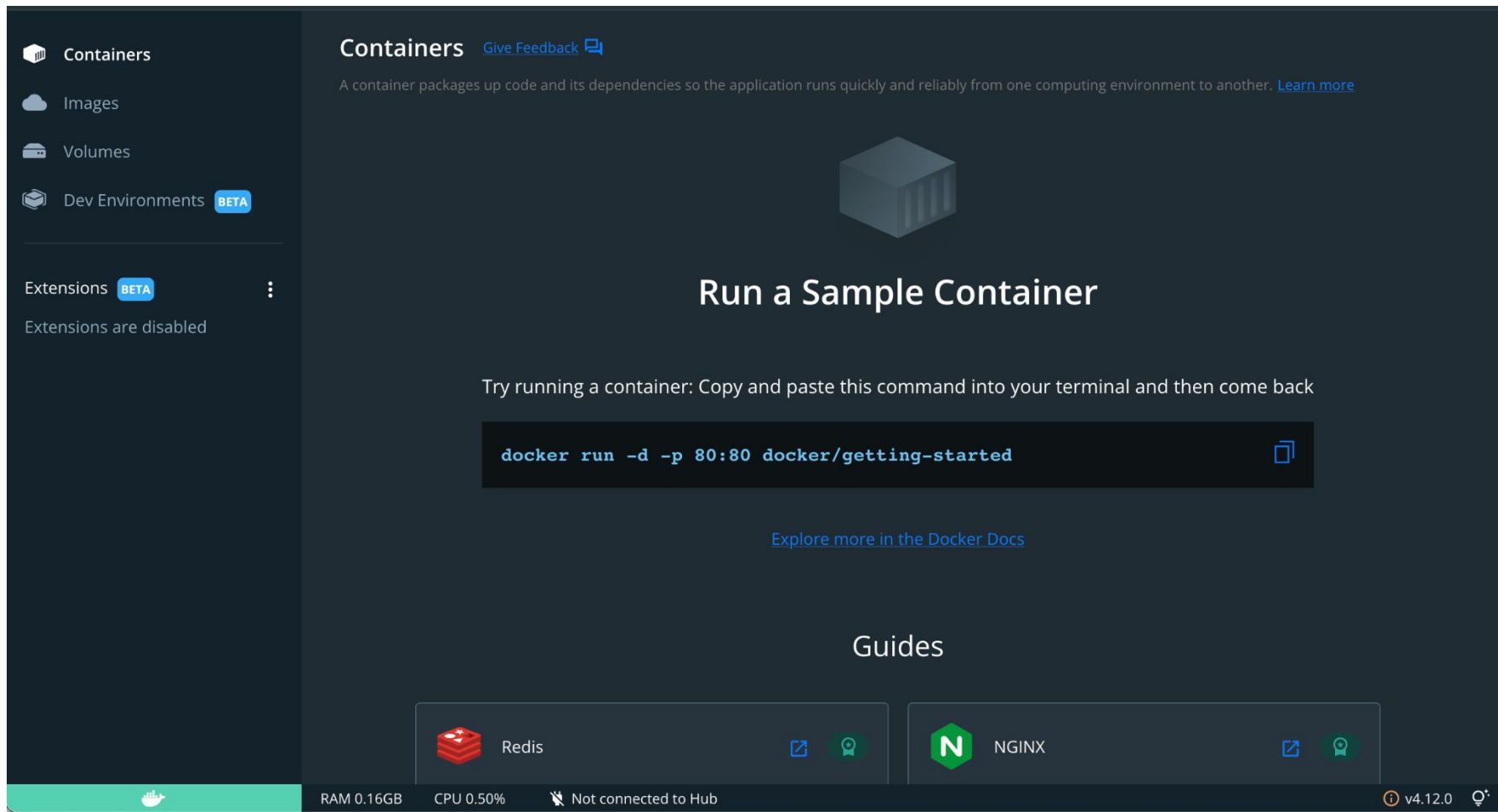
With <https://polarlightning.com>



# Install Docker Desktop and Polar

- [Docker Desktop](#)
- [Polar](#)

# 1. Open Docker Desktop, wait for it to start



The screenshot shows the Docker Desktop application window. On the left is a dark sidebar with navigation options: 'Containers' (selected), 'Images', 'Volumes', 'Dev Environments' (with a 'BETA' badge), 'Extensions' (with a 'BETA' badge), and a status message 'Extensions are disabled'. The main area has a dark background. At the top, it says 'Containers' with a 'Give Feedback' link. Below that is a descriptive sentence about containers and a 'Learn more' link. In the center is a large 3D cube icon and the heading 'Run a Sample Container'. Below this is a text prompt: 'Try running a container: Copy and paste this command into your terminal and then come back'. A dark box contains the command `docker run -d -p 80:80 docker/getting-started` with a copy icon. Below the box is a link 'Explore more in the Docker Docs'. At the bottom of the main area is a 'Guides' section. The very bottom of the window is a status bar showing system resources: 'RAM 0.16GB', 'CPU 0.50%', and a connection status 'Not connected to Hub'. On the right side of the status bar are version information 'v4.12.0' and a user profile icon. Two container cards are visible at the bottom: 'Redis' with a red cube icon and 'NGINX' with a green 'N' icon. Each card has a link icon and a status icon.

Containers [Give Feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

## Run a Sample Container

Try running a container: Copy and paste this command into your terminal and then come back

```
docker run -d -p 80:80 docker/getting-started
```

[Explore more in the Docker Docs](#)

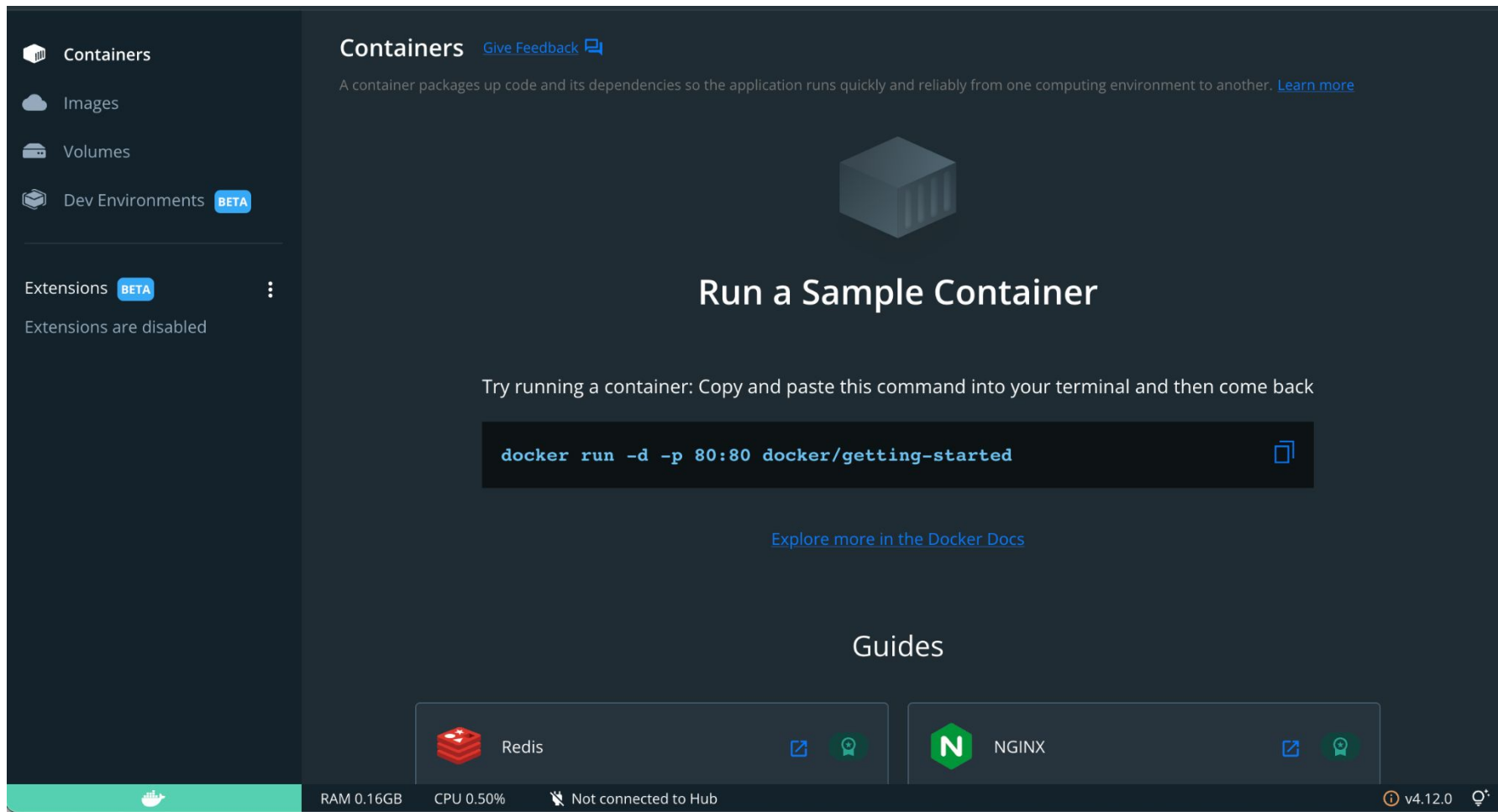
### Guides

Redis

NGINX

RAM 0.16GB CPU 0.50% Not connected to Hub v4.12.0

# 1. Open Docker Desktop, wait for it to start



The screenshot shows the Docker Desktop application window. On the left is a dark sidebar with navigation options: Containers (selected), Images, Volumes, Dev Environments (with a BETA badge), Extensions (with a BETA badge), and a status message 'Extensions are disabled'. The main area has a header 'Containers' with a 'Give Feedback' link. Below this is a descriptive sentence about containers and a 'Learn more' link. A large 3D cube icon is centered. The main heading is 'Run a Sample Container'. Below it, text instructs the user to run a command in a terminal. A dark box contains the command: `docker run -d -p 80:80 docker/getting-started`, with a copy icon to its right. Below the command box is a link to 'Explore more in the Docker Docs'. At the bottom, there's a 'Guides' section with two cards: 'Redis' and 'NGINX', each with an icon, a name, and two small action icons. The very bottom status bar shows system metrics: 'RAM 0.16GB', 'CPU 0.50%', and a connection status 'Not connected to Hub' with a warning icon. On the far right of the status bar, it shows the version 'v4.12.0' and a user profile icon.

Containers [Give Feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)




## Run a Sample Container




Try running a container: Copy and paste this command into your terminal and then come back



```
docker run -d -p 80:80 docker/getting-started
```

[Explore more in the Docker Docs](#)

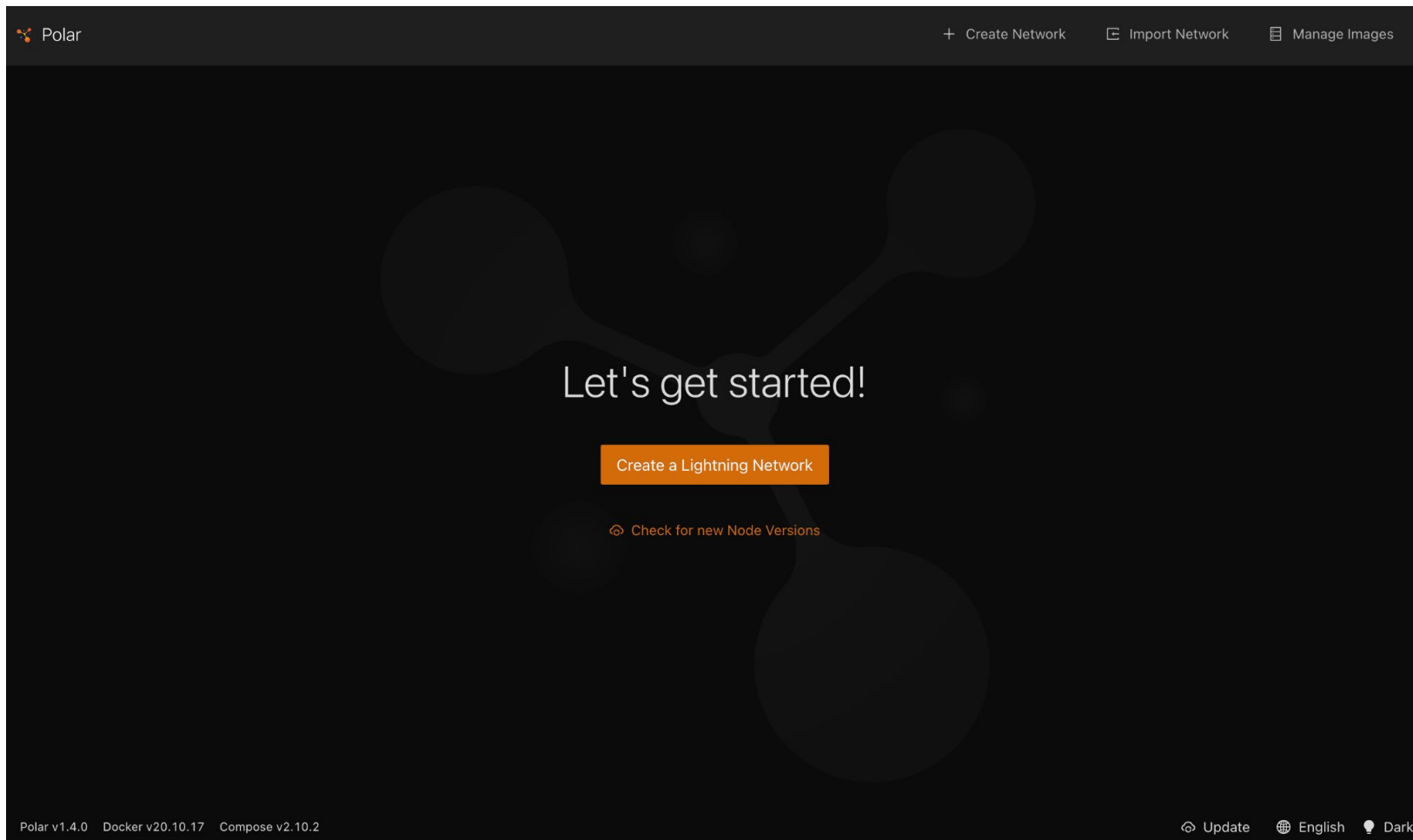
### Guides

 Redis  

 NGINX  

RAM 0.16GB CPU 0.50%  Not connected to Hub v4.12.0 

## 2. Open Polar, Create a Lightning network





### 3. Create your network

- Give it a name
- You can use the default network or customize it / add nodes
- Make sure that you have at least 1 Bitcoin Core node though

#### ← Create a new Lightning Network

\* Network Name

Bitcoinplebdev

How many Managed Nodes?

\* LND

1

\* Core Lightning

1

\* Eclair

1

\* Bitcoin Core

1

Create Network

# 4. Click Start

Polar

+ Create Network    Import Network    Manage Images

← Bitcoinplebdev Started Stop ⋮

```
graph TD; alice --- bob; bob --- carol; bob --- backend1; alice --- backend1; carol --- backend1;
```

**Network Designer**

Click on an element in the designer to see details

**Add Nodes**

Drag a node below onto the canvas to add it to the network

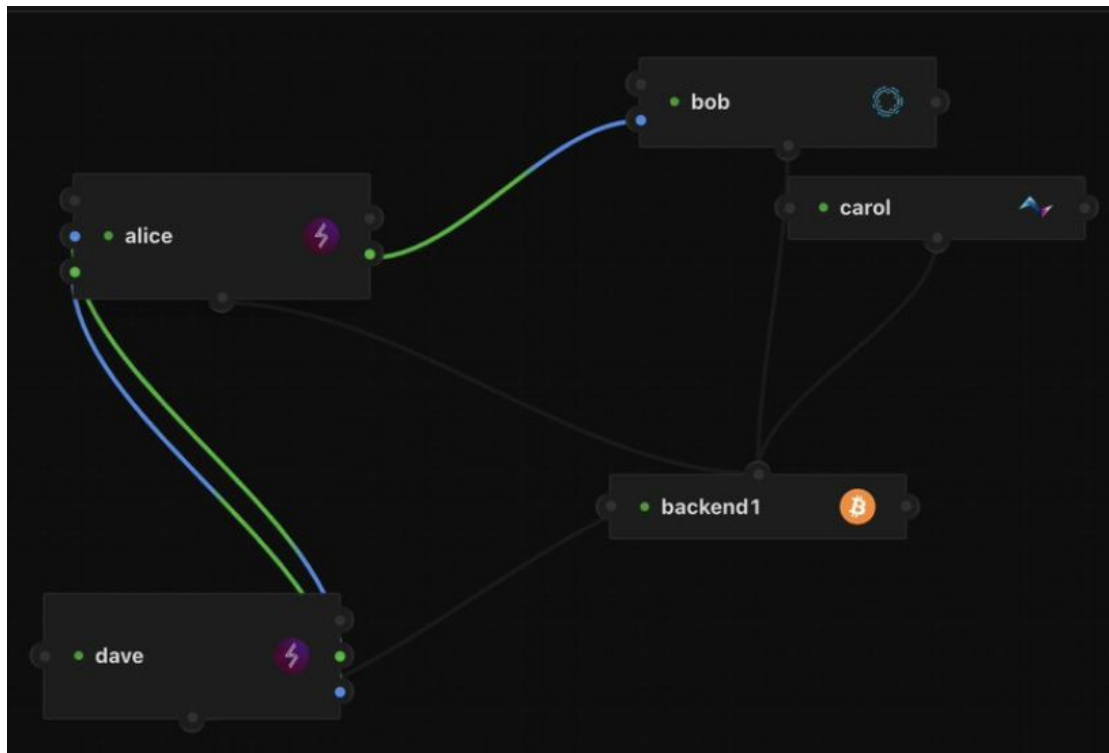
Show All Versions ☒

- LND v0.15.2-beta<sup>latest</sup>
- LND v0.15.1-beta
- LND v0.15.0-beta
- LND v0.14.3-beta
- LND v0.13.1-beta

Polar v1.4.0    Docker v20.10.17    Compose v2.10.2

Update    English    Dark

# Let's create some channels / invoices in Polar



# Resources

- Bitcoin's Lightning Network, Simply Explained! - [https://www.youtube.com/watch?v=rrr\\_zPmEiME](https://www.youtube.com/watch?v=rrr_zPmEiME)
- A Technical Introduction to The Lightning Network - [https://www.youtube.com/watch?v=E1n3sKKPD\\_k&t=330s](https://www.youtube.com/watch?v=E1n3sKKPD_k&t=330s)
- The Lightning whitepaper - <https://lightning.network/lightning-network-paper.pdf>
- Lightning Series: Mastering Lightning with Andreas M. Antonopoulos & René Pickhardt - <https://www.youtube.com/watch?v=zG8PZsHLung>
- Understanding the Lightning Network (series) - <https://bitcoinmagazine.com/technical/understanding-the-lightning-network-part-building-a-bidirectional-payment-channel-1464710791>
- Mastering the Lightning Network Book (free) - <https://github.com/lnbook/lnbook>