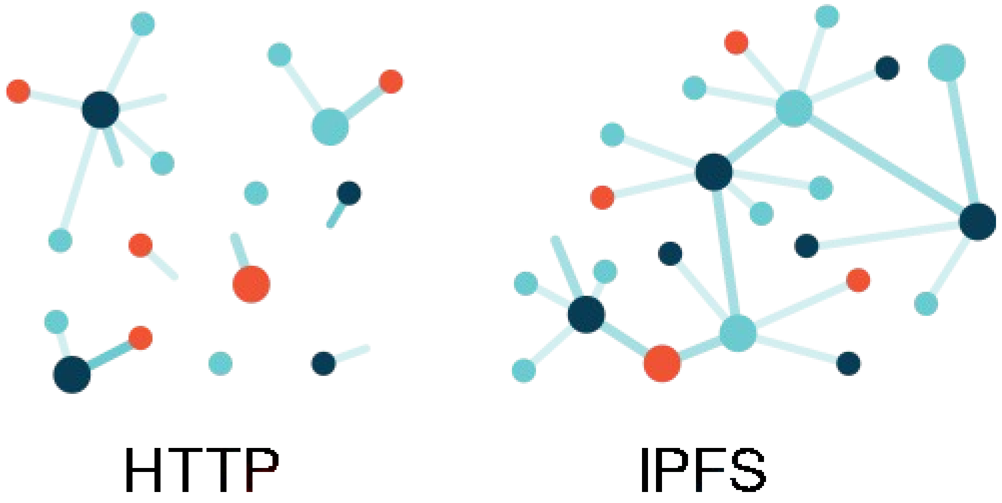
# IPFS – The Interplanetary File System

William Harrell
LUG @ NC State
October 2nd, 2019

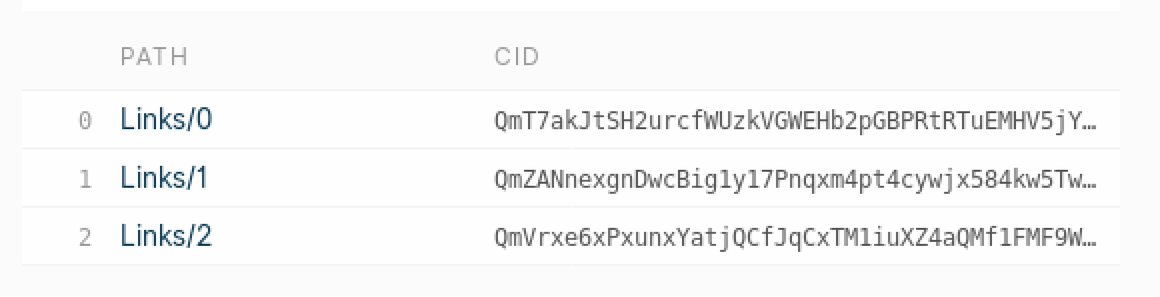
# Basics

- IPFS is a peer-to-peer network that aims to do everything HTTP can in a more distributed way
- Data is uploaded to the network and broken into blocks
- More popular blocks are requested more, and thus propagate through greater portions of the network
- Blocks are identified by the hash of their contents



# Content Identifiers (CIDs)

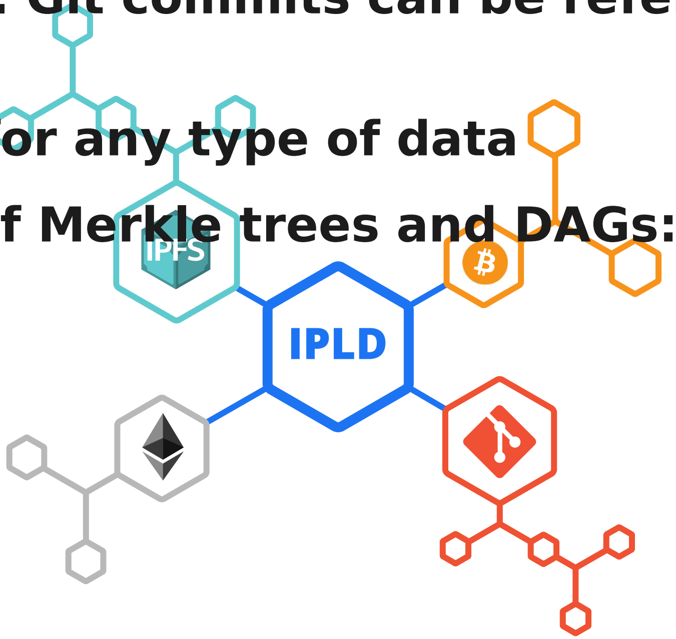
- In most file systems, data is retrieved using location
  - “Get me the file at /home/william/Desktop/yeet.png”
  - “Get me the file at www.google.com”
- With IPFS, data is retrieved using the hash of its contents
  - “Get me the file whose data hashes to QmWge3mzMBerFt7ZUN9ds2skLvC3ejLkUST4oPAZWwhN7A”
- All files and directories are given a hash; no prefix or otherwise obvious organization (usually, can be given human readable names)



|   | PATH | CID |
|---|---|---|
| 0 | Links/0 | QmT7akJtSH2urcfWUzkVGWEHb2pGBPRtRTuEMHV5jY… |
| 1 | Links/1 | QmZANnexgnDwcBig1y17Pnqxm4pt4cywjx584kw5Tw… |
| 2 | Links/2 | QmVrxe6xPxunxYatjQCfJqCxTM1iuXZ4aQMf1FMF9W… |

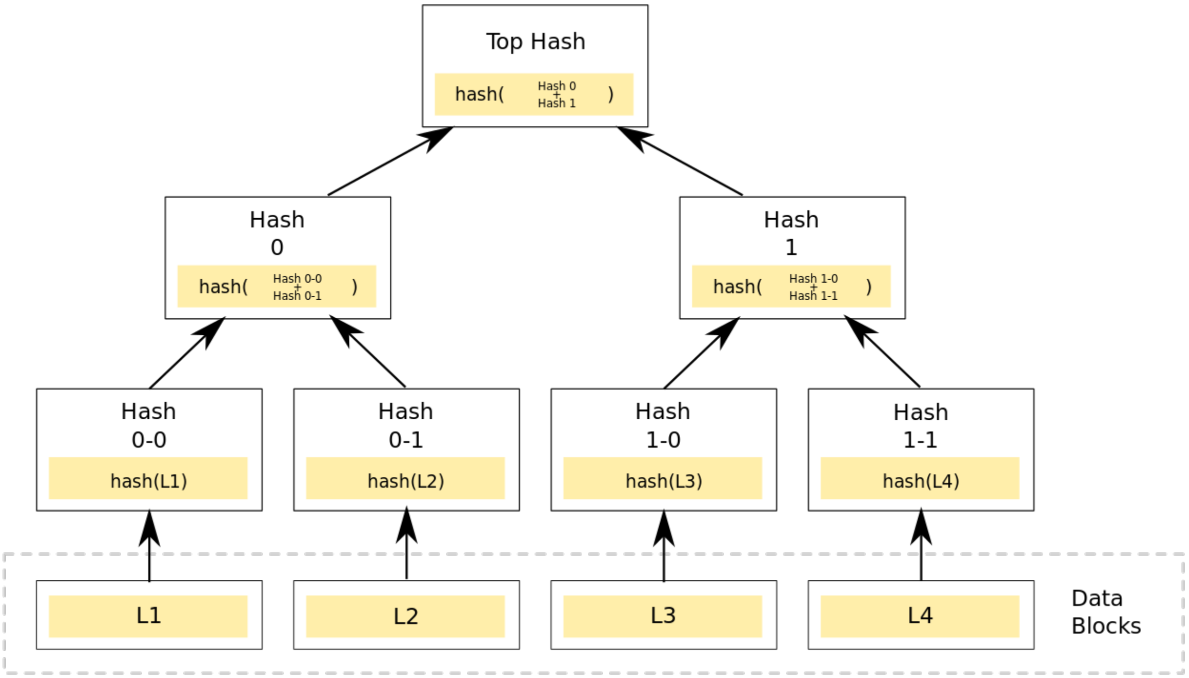
# Interplanetary Linked Data (IPLD)

- **If data is broken into blocks, how is the entire file reconstituted? How are directories stored?**
- **IPLD is a set of specifications, separate from the IPFS project, that allows for hash identified content of all protocols to be linked together**
- **Not just for IPFS (e.g. Git commits can be referenced in Bitcoin transactions)**
- **IPLD is hyperlinking for any type of data**
- **Uses a combination of Merkle trees and DAGs: the MerkleDAG**



# Merkle Tree

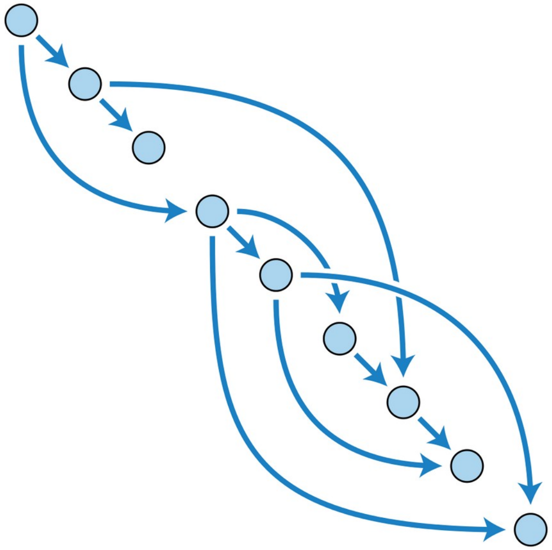
- **Binary tree with n-nodes**
- **Leaves contain data and a hash**
- **Non-leaves contain a hash and references to hashed children**



Top Hash
hash( Hash 0 + Hash 1 )
Hash 0
hash( Hash 0-0 + Hash 0-1 )
Hash 1
hash( Hash 1-0 + Hash 1-1 )
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4
Data Blocks

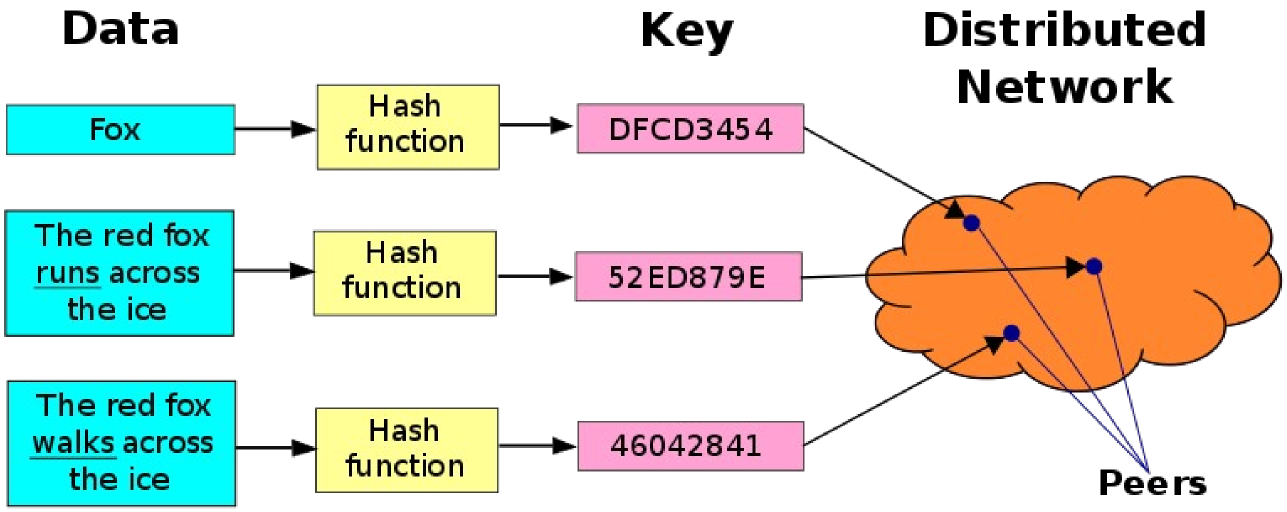
# Directed Acyclic Graph (DAG)

- Type of graph with two very important properties
  - Direction: References between nodes are one-way
    - Data cannot contain itself!
  - Acyclic: Loops cannot be created within the network
    - A child folder cannot contain its parent, or its parent’s parent, or its ...!



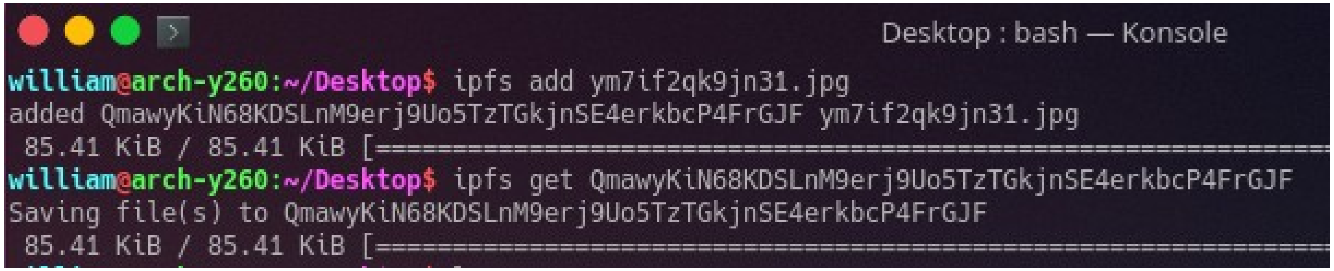
# Distributed Hash Tables (DHTs)

- **Hash table in which hashes map to blocks of data**
- **Also used by other P2P protocols such as BitTorrent**
- **Someone queries the DHT for keys; they are directed to the peer that has the matching block**
- **In IPFS, values can point to nodes in a MerkleDAG to represent larger files and folders**

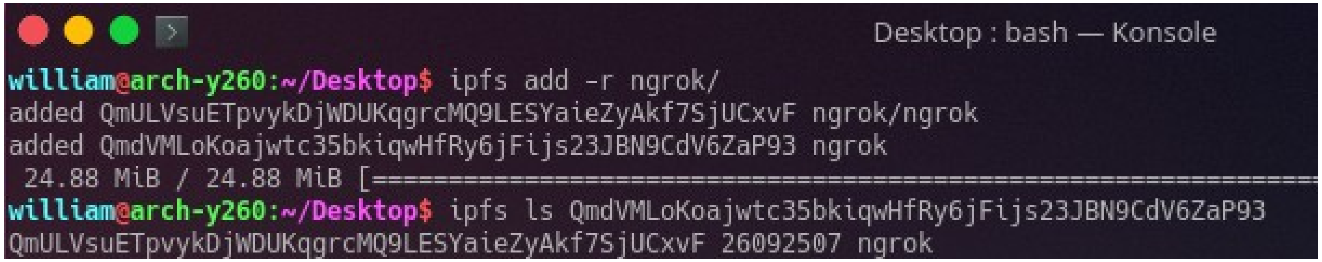


# Go CLI

- **ipfs add <path>: Adds a local file to IPFS**
- **ipfs cat/get <ipfs path>: Download an IPFS reference and reconstruct it**
- **ipfs ls <ipfs path>: List the files/directories in a reference**
- **Many more commands, these are just the basics**



```
william@arch-y260:~/Desktop$ ipfs add ym7if2qk9jn31.jpg
added QmawyKiN68KDSLnM9erj9Uo5TzTGkjnSE4erkbcP4FrGJF ym7if2qk9jn31.jpg
 85.41 KiB / 85.41 KiB [==========================================================
william@arch-y260:~/Desktop$ ipfs get QmawyKiN68KDSLnM9erj9Uo5TzTGkjnSE4erkbcP4FrGJF
Saving file(s) to QmawyKiN68KDSLnM9erj9Uo5TzTGkjnSE4erkbcP4FrGJF
 85.41 KiB / 85.41 KiB [==========================================================
```



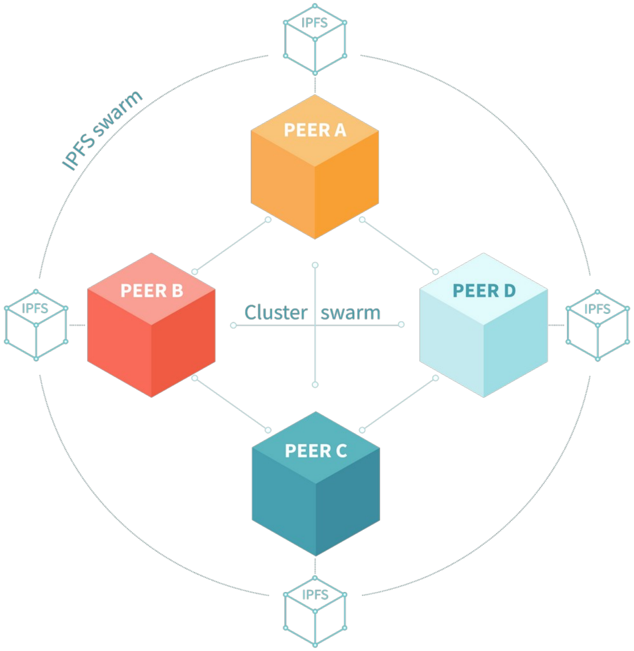
```
william@arch-y260:~/Desktop$ ipfs add -r ngrok/
added QmULVsuETpvykDjWDUKqgrcMQ9LESYaieZyAkf7SjUCxvF ngrok/ngrok
added QmdVMLoKoajwtc35bkiqwHfRy6jFijs23JBN9CdV6ZaP93 ngrok
 24.88 MiB / 24.88 MiB [==========================================================
william@arch-y260:~/Desktop$ ipfs ls QmdVMLoKoajwtc35bkiqwHfRy6jFijs23JBN9CdV6ZaP93
QmULVsuETpvykDjWDUKqgrcMQ9LESYaieZyAkf7SjUCxvF 26092507 ngrok
```

# HTTP Gateways

- As cool as IPFS is, your non-Linux-using friends and/or parents probably aren’t going to set it up
- HTTP gateways exist to allow retrieval of IPFS resources from HTTP
- The official one is gateway.ipfs.io/ipfs or /ipns (more on that later)
- Requesting a file through the gateway still makes the file look popular like any other request, resource is cached but not pinned

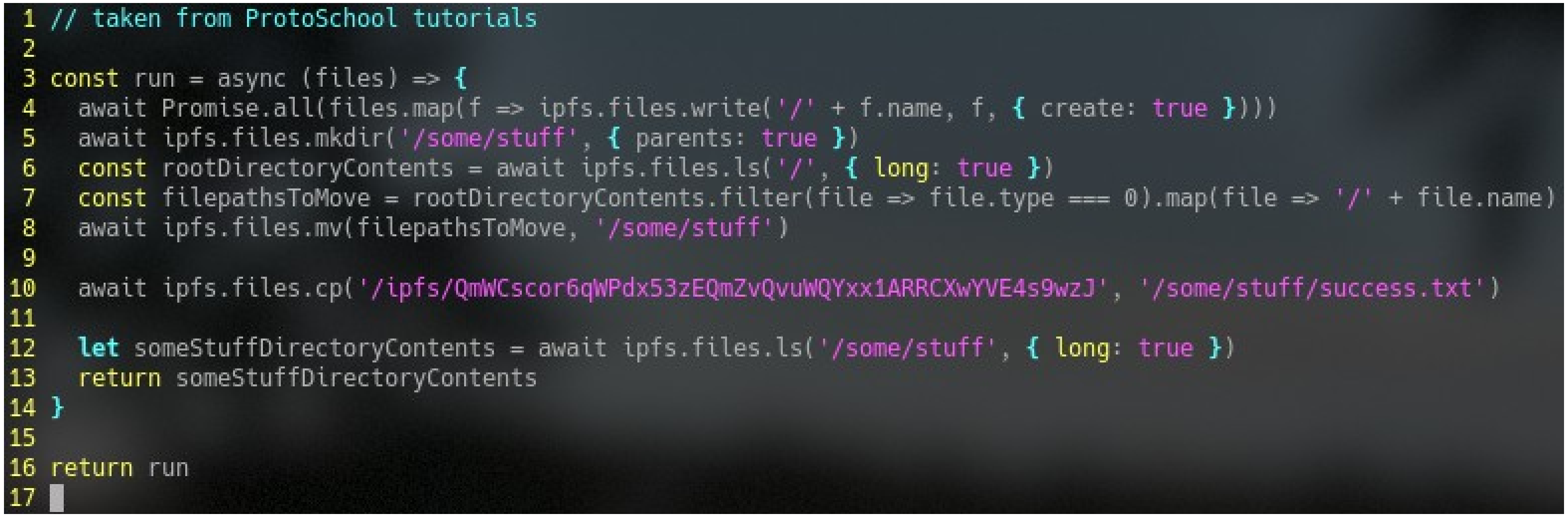
# Pinning and IPFS Cluster

- When a resource is requested, it will be cached but this is only temporary
- Pinning a resource will make it permanently available
- IPFS Cluster manages multiple nodes and splits blocks corresponding to pins across the nodes, rather than keeping them all on one node



# js-ipfs

- **Run an IPFS node in a browser tab, without the need for IPFS to be running on the computer**
- **Implements same functionality as the Go CLI**



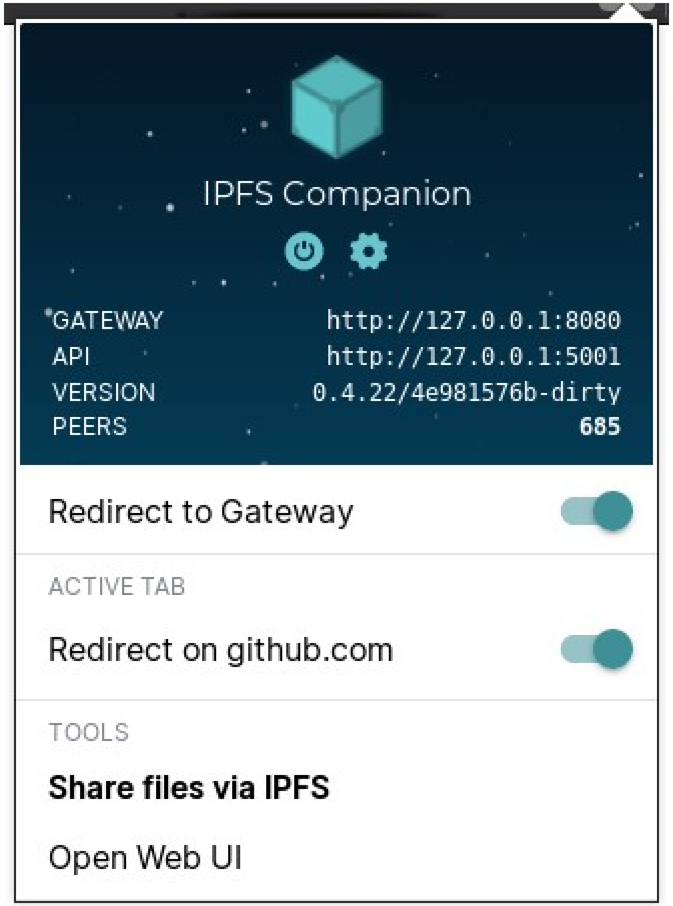
```
// taken from ProtoSchool tutorials

const run = async (files) => {
  await Promise.all(files.map(f => ipfs.files.write('/' + f.name, f, { create: true })))
  await ipfs.files.mkdir('/some/stuff', { parents: true })
  const rootDirectoryContents = await ipfs.files.ls('/', { long: true })
  const filepathsToMove = rootDirectoryContents.filter(file => file.type === 0).map(file => '/' + file.name)
  await ipfs.files.mv(filepathsToMove, '/some/stuff')

  await ipfs.files.cp('/ipfs/QmWCscor6qWPdx53zEQmZvQvuWQYxx1ARRCXwYVE4s9wzJ', '/some/stuff/success.txt')

  let someStuffDirectoryContents = await ipfs.files.ls('/some/stuff', { long: true })
  return someStuffDirectoryContents
}

return run

```

# IPNS and DNSLink

- Two problems with content addressing
  - Hashes are not human readable
  - Changing the content of the website’s files gives a new identifer
- IPNS solves second problem by pointing your public key to a reference
  - Still unreadable
  - Also slow; devs don’t recommend using it
- DNSLink solves both problems but requires a domain
  - DNS TXT record created on your domain containing an IPFS reference

# Browser Companion

- Redirects IPFS links to local gateway
- Auto detect websites with DNSLink entries, load them over IPFS instead of HTTP
- Access to local node in JavaScript through window.ipfs object



IPFS Companion

```
GATEWAY        http://127.0.0.1:8080
API            http://127.0.0.1:5001
VERSION    0.4.22/4e981576b-dirty
PEERS                         685
```

Redirect to Gateway

ACTIVE TAB

Redirect on github.com

TOOLS

**Share files via IPFS**

Open Web UI

# Web UI

- View statistics
- Manage files
- Explore IPFS objects
- Opened with localhost:5001/webui

**Images taken shamelessly (but legally) from IPFS.io and Wikimedia**