Getting Started

If you haven't done so, install ipfs.

init the repo

`ipfs` uses a global local object repository, added to `~/.ipfs`:

```
> ipfs init
initializing ipfs node at /Users/jbenet/.go-ipfs
generating 4096-bit RSA keypair...done
peer identity: Qmcpo2iLBikrdf1d6QU6vXuNb6P7hwrbNPW9kLAH8eG67z
to get started, enter:

  ipfs cat /ipfs/QmPXME1oRtoT627YKaDPDQ3PwA8tdP9rWuAAweLzqSwAWT/readme
```

Note the hash there may differ. If it does, use the one you got.

Now, try running:

```
ipfs cat /ipfs/QmPXME1oRtoT627YKaDPDQ3PwA8tdP9rWuAAweLzqSwAWT/readme
```

You should see something like this:

```
Hello and Welcome to IPFS!

██╗██████╗ ███████╗███████╗
██║██╔══██╗██╔════╝██╔════╝
██║██████╔╝█████╗  ███████╗
██║██╔═══╝ ██╔══╝  ╚════██║
██║██║     ██║     ███████║
╚═╝╚═╝     ╚═╝     ╚══════╝

If you're seeing this, you have successfully installed
IPFS and are now interfacing with the ipfs merkledag!

 -------------------------------------------------------
| Warning:                                              |
|   This is alpha software. use at your own discretion! |
|   Much is missing or lacking polish. There are bugs.  |
|   Not yet secure. Read the security notes for more.   |
 -------------------------------------------------------

Check out some of the other files in this directory:
```

```
  ./about
  ./help
  ./quick-start     <-- usage examples
  ./readme          <-- this file
  ./security-notes
```

You can explore other objects in there. In particular, check out `quick-start`:

```
ipfs cat /ipfs/QmPXME1oRtoT627YKaDPDQ3PwA8tdP9rWuAAweLzqSwAWT/quick-start
```

Which will walk you through several interesting examples.

Going Online

Once you're ready to take things online, run the daemon in another terminal:

```
> ipfs daemon
Initializing daemon...
API server listening on /ip4/127.0.0.1/tcp/5001
Gateway server listening on /ip4/127.0.0.1/tcp/8080
```

Wait for all three lines to appear.

> Make note of the tcp ports you get. if they are different, use yours in the commands below.

Now, if you're connected to the network, you should be able to see the ipfs addresses of your peers:

```
> ipfs swarm peers
/ip4/104.131.131.82/tcp/4001/ipfs/QmaCpDMGvV2BGHeYERUEnRQAwe3N8SzbUtfsmvsqQLuvuJ
/ip4/104.236.151.122/tcp/4001/ipfs/QmSoLju6m7xTh3DuokvT3886QRYqxAzb1kShaanJgW36yx
/ip4/134.121.64.93/tcp/1035/ipfs/QmWHyrPWQnsz1wxHR219ooJDYTvxJPyZuDUPSDpdsAovN5
/ip4/178.62.8.190/tcp/4002/ipfs/QmdXzZ25cyzSF99csCQmmPZ1NTbWTe8qtKFaZKpZQPdTFB
```

These are a combination of `<transport address>/ipfs/<hash-of-public-key>`.

Now, you should be able to get objects from the network. Try:

```
ipfs cat /ipfs/QmW2WQi7j6c7UgJTarActp7tDNikE4B2qXtFCfLPdsgaTQ/cat.jpg >cat.jpg
open cat.jpg
```

And, you should be able to give the network objects. Try adding one, and then viewing it in your favorite brower:

```
> hash=`echo "I <3 IPFS -"\`whoami\` | ipfs add -q`
> curl "http://ipfs.io/ipfs/$hash"
I <3 IPFS -<your username>
```

Cool, huh? The gateway served a file *from your computer*. The gateway queried the DHT, found your machine, requested the file, your machine sent it to the gateway, and the gateway sent it to your browser.

Note: depending on the state of the network, the `curl` may take a while. The public gateways may be overloaded or having a hard time reaching you.

You can also check it out at your own local gateway:

```
> curl "http://127.0.0.1:8080/ipfs/$hash"
I <3 IPFS -<your username>
```

By default, your gateway is not exposed to the world, it only works locally.

Fancy Web Console

We also have a web console you can use to check the state of your node. On your favorite web browser, go to:

> http://localhost:5001/webui

This should bring up a console like this:

Now, you're ready:

Onward to more Examples →