Course Project: Exploring IPFS

Introduction

In this project, you will explore Interplanetary File System (IPFS) and some of the commands for operating on IPFS. IPFS is a global, versioned, content addressed peer-to-peer file system. Though we covered a diverse set of topics in this course, we have chosen IPFS as the topic for the course project due to its immediate applicability to the design of blockchain solutions. Recall from the courses 1-3 that blockchain is not a database/store. Typically the metadata in an application's data and the timeline of transactions are stored on the distributed ledger of the blockchain. The application data itself can be stored in a distributed database or a decentralized database such as the IPFS.

Learning Outcomes

On completion of this project, you will be able to

- Install IPFS on your compute machine
- Use command line interface (CLI) to IPFS to add a file on IPFS
- Use the web interface to the IPFS to view your peers and add (drag and drop) files to IPFS
- Add a folder of files to IPFS using the web interface to the IPFS, view the hash of the folder and the files on the CLI
- Use IPFS to share files with others
- Explain the base-58 addressing scheme of IPFS objects

What to do?

You will install the IPFS software and follow the instructions given below to complete the project steps described.

After completing the steps, answer the summative quiz associated with the Module 4 of the Course 4.

Installation

1. To download ipfs visit https://dist.ipfs.io/#go-ipfs and download go-ipfs for your platform (Version: v0.4.14 or above). This is a Go language implementation of the IPFS. Note that there is also a javascript implementation of IPFS: js-ipfs.
   Go-ipfs implementation of IPFS includes IPFS core, daemon server, CLI, HTTP API and a HTTP Gateway for easy access to the IPFS.
2. After downloading, do the following
   a. **Mac/Linux/VM from Course 1:**
      ```
      tar xvfz <go-ipfs distribution>.tar.gz
      cd go-ipfs
      ./install.sh
      ```
      Basically install.sh will copy the ipfs executable into the bin folder.

   b. **Windows:** You can extract (**Extrall All**) into Course4 directory, if you have created one.

NOTE: Linux/Mac users can use ipfs commands from any directory.
However, Windows users have to use the ipfs commands from the go-ipfs directory. This can be overcome by using an environment variable by setting ipfs = <location of go-ipfs>, however, we DO NOT advise that as any unintentional, accidental changes in the original environment variables may cause the learner problems. Only change the variables if you are sure about it.

Initialization

1. Next initialize the ipfs repository; execute the command **ipfs init**
2. You should be able to see,
   a. Location where IPFS node was initialized
   b. Completion of 2048-bit RSA keypair generation
   c. Your peer identity
   d. See below for a sample output from my system.
   e. A command to get started:
      ipfs cat /ipfs/QmS4ustL54uo8FzR9455qaxZwuMiUhyvMcX9Ba8nUH4uVv/readme

The hash value you get will be different. Execute the command with the hash you get and see what happens.

```
C:\Users\bina\Documents\Course4\go-ipfs>ipfs init
initializing IPFS node at C:\Users\bina\.ipfs
generating 2048-bit RSA keypair...done
peer identity: QmZaszUohueQ6q4LQEkhP3AUwpLK4Zx7Gg2N7LuZbCNuv2
to get started, enter:

    ipfs cat /ipfs/QmS4ustL54uo8FzR9455qaxZwuMiUhyvMcX9Ba8nUH4uVv/readme
```

Things to observe: The IPFS address is made of base-58 alphanumeric characters. So is the Bitcoin address. A 256-bit key can be expressed as a hexadecimal string with 58 characters and 44 characters in Base58. The leading "Qm" includes representation for the encoding scheme of SHA-256. We will not go into details of the algorithms. However it is significant to recognize and understand Base58 since it is commonly used in crypto-world. Base58 is explained below:

26 Capital letters minus (O, I) = 24 capital letters
26 Lowercase letters minus (l) = 25 lowercase letters
10 numerals minus (0) = 9 numerals
Total = 58, thus base-58.

With a length of 44 characters and 58 available characters for every position, we get a total of 58^{44} possible combinations for the strings which is almost equivalent to $3.89 * 10^{77}$. To put that in perspective, the total number of atoms in the observable universe is considered to be around 7.1×10^{79}[1]. With this scheme, you can name 0.5% of all the atoms in the observable universe with a few names to spare!

Running an example

1. Copy the three files provided in the project directory: **pet.jpg, myfile1.txt and myfile2.txt** to your desktop.
2. Copy **myfile1.txt** from the desktop to your go-ipfs folder.
3. Execute the command **ipfs add myfile1.txt** for adding a file to IPFS. You will use **ipfs add -r <your directory>** to add a directory to IPFS.
4. The command will return a hash value which is used as a unique identifier for the file. Anyone having the hash will be able to access the file
5. You can view the contents of the hash locally to double check, using the **ipfs cat <the-hash-you-got-here>** command
6. Open a new console/terminal and execute **ipfs daemon &.** This will spawn the ipfs daemon.
   This is required for the ipfs gateway to interact with your file system and this allows others with the hash to access the file. & keeps the daemon in the background allowing you to work with CLI.
7. You can also view the file using the ipfs gateway, copy paste the url in your browser after pasting the hash https://gateway.ipfs.io/ipfs/**<hash of myfile1.txt>.** Be patient. Depending on your computer/laptop power it may take some time for the initial access to the IPFS from the web.
8. The above link can be shared with anyone, even people who haven't configured IPFS. It will behave like a regular link.
9. Now an **"ls"** command form the current go-ipfs directory will list both the unix file as well as the IPFS hash of the file we added. Try this command.
10. Let's now move into the web console. It is easy to add files using the web console to the IPFS system. Make sure you had started the daemon as indicated in step 8 above.

Web console:

1. Since you have already started the daemon we can interact with the help of the browser rather than using command line interface. You can drag and drop files.
2. Open any browser and go to **localhost:5001/webui** or **127.0.0.1:5001/webui**
3. All command line actions that we discussed can be accomplished using the web interface. You can play around with uploading files and sharing hashes.
4. Be patient. Depending on your computer/laptop power it may take some time for the initial access to the IPFS from the web.
5. In the screen shot show examine (i) peer id, and (ii) the tabs on the left panel: Home, Connections, Files, DAG, and Config.
6. Click on **DAG** and add the hash of the myfile1.txt, you should be able to see the content of the text file.
7. When you click on the **Connections** tab you will be able to see all the peers that are connected you in the global IPFS system. You will see a globe that is constantly updated with newer connections.
8. Now navigate to the **Files** tab, click on it. You will see file explorer interface in the center for adding objects to your IPFS. Create a new folder **project4** by clicking on the Create Folder command.
9. Upload the three files, **myfile1.txt**, **myfile2.txt** and **pet.jpg** into this folder by dragging and dropping them from the desktop into this folder. (Recall you copied them earlier in Step 1 of the previous exercise.). This will add the folder as well as files into your IPFS.
10. If you right click on any of the items to copy the hash address of the item.
11. Now get back the CLI (Command Line Interface) and examine if you can see the new additions to the file system. Execute the command **ipfs files ls –l**
    You should see the newly added files, their hash and their size.
12. You can get the newly added file and also view the content**. ipfs get <hash of newly added>**
13. This ipfs hash is universally accessible by anyone including a friend of yours who has configured the IPFS system as described above. All you have to do is send the hash of the folder or file you want to share. Warning: this may take some time depending on network configuration at both ends of transfer as well the size of the files.
14. People who haven’t configured the IPFS system can also view the contents on https://gateway.ipfs.io/<hash value>. Please be patient as this may take some time.

Commands:

Ipfs init	:	initializes ipfs in your system(need to run only once)
ipfs id	:	Show ipfs node id info
ipfs cat <ipfs_path/ref>	:	Displays the IPFS object data from the path
ipfs add <file_name>	:	Adds a file to IPFS
ipfs add -r <directory_name>	:	Adds all the files in the directory
ipfs get <ref>	:	Downloads the IPFS objects
ipfs ls <ref>	:	Lists the links from an object
ipfs refs <ref>	:	Lists the hashes of links from an object
ipfs pin add <ipfs-path>	:	Pin objects to local storage
ipfs pin rm <ipfs-path>	:	Remove pinned objects from local storage
ipfs pin ls	:	Lists all the objects of the files/folder pinned to your repo
ipfs daemon	:	Run a network-connected IPFS node
ipfs version	:	Returns current version of ipfs
ipfs files ls	:	List directories in the local mutable namespace.

Reference:

1. Installation: https://ipfs.io/docs/getting-started/
2. Commands: https://ipfs.io/docs/commands/#ipfs-cat
3. IPFS global namespace: https://gateway.ipfs.io/
4. Number of atoms in the universe: https://physics.stackexchange.com/questions/47941/dumbed-down-explanation-how-scientists-know-the-number-of-atoms-in-the-universe

Troubleshooting:
During the testing of IPFS if you cannot reinitialize ipfs again, it would through the following error,
initializing IPFS node at <$PATH>/.ipfs
Error: ipfs configuration file already exists!

Be warned that reinitializing would overwrite your keys and you may not want to do this in a live system.

You need to delete <$PATH>/.ipfs, **rm -rf <$PATH>/.ipfs** and then try **ipfs init**
You can use the same command above if you don't want to use IPFS anymore.