IPFS ON AWS

Rahul Raj Mogili - 1900425
Nirupam Bidikar - 1878058
Pranav Saineni - 1884587

Overview

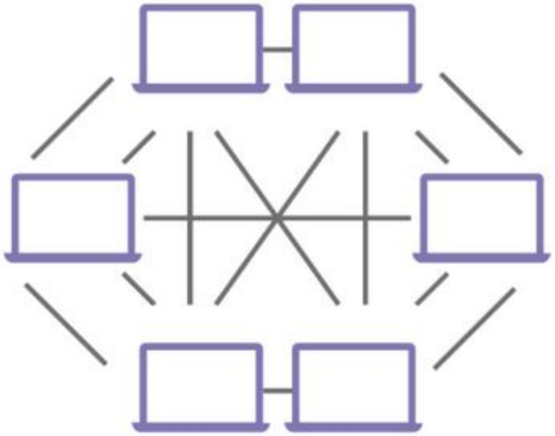
- Approaches
- Goals we achieved
- Setbacks
- Future Prospects
- References

Approach-1

- Creating a private IPFS network
- Using HTTP API exposed by nodes.
- Structure of the base API

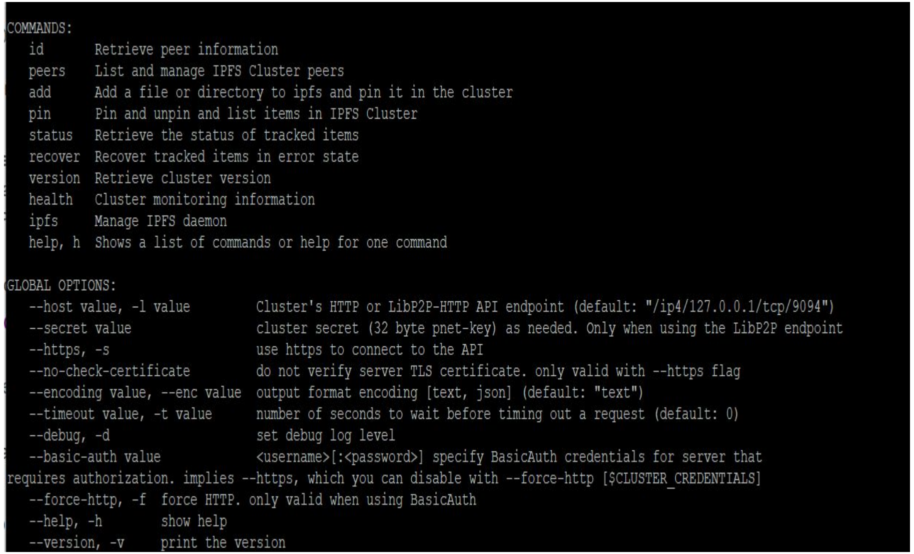
http://<node-ip>/api/v0/<function>

- Can perform all the commands carried out in the CLI.
- Insecure as anybody with the IP can interact with the node.



Approach -1 contd

- Files added in this approach would have to be manually replicated across network.
- Can be automated through scripts
- Manual replication is complicated as we have to deal with hashes.

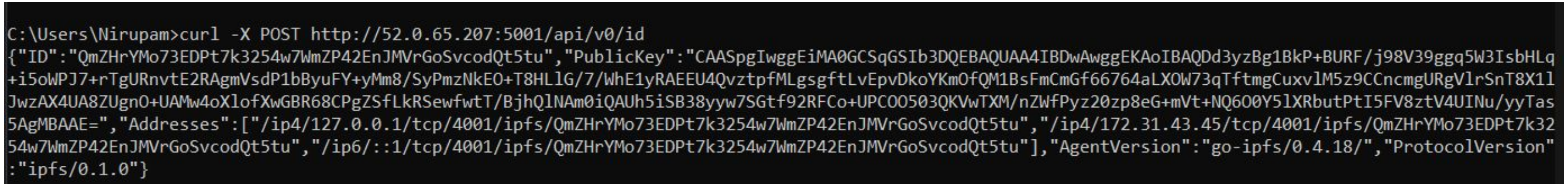


```
COMMANDS:
   id       Retrieve peer information
   peers    List and manage IPFS Cluster peers
   add      Add a file or directory to ipfs and pin it in the cluster
   pin      Pin and unpin and list items in IPFS Cluster
   status   Retrieve the status of tracked items
   recover  Recover tracked items in error state
   version  Retrieve cluster version
   health   Cluster monitoring information
   ipfs     Manage IPFS daemon
   help, h  Shows a list of commands or help for one command

GLOBAL OPTIONS:
   --host value, -l value        Cluster's HTTP or LibP2P-HTTP API endpoint (default: "/ip4/127.0.0.1/tcp/9094")
   --secret value                cluster secret (32 byte pnet-key) as needed. Only when using the LibP2P endpoint
   --https, -s                   use https to connect to the API
   --no-check-certificate        do not verify server TLS certificate. only valid with --https flag
   --encoding value, --enc value output format encoding [text, json] (default: "text")
   --timeout value, -t value     number of seconds to wait before timing out a request (default: 0)
   --debug, -d                   set debug log level
   --basic-auth value            <username>[:<password>] specify BasicAuth credentials for server that
requires authorization. implies --https, which you can disable with --force-http [$CLUSTER_CREDENTIALS]
   --force-http, -f  force HTTP. only valid when using BasicAuth
   --help, -h        show help
   --version, -v     print the version
```

HTTP API

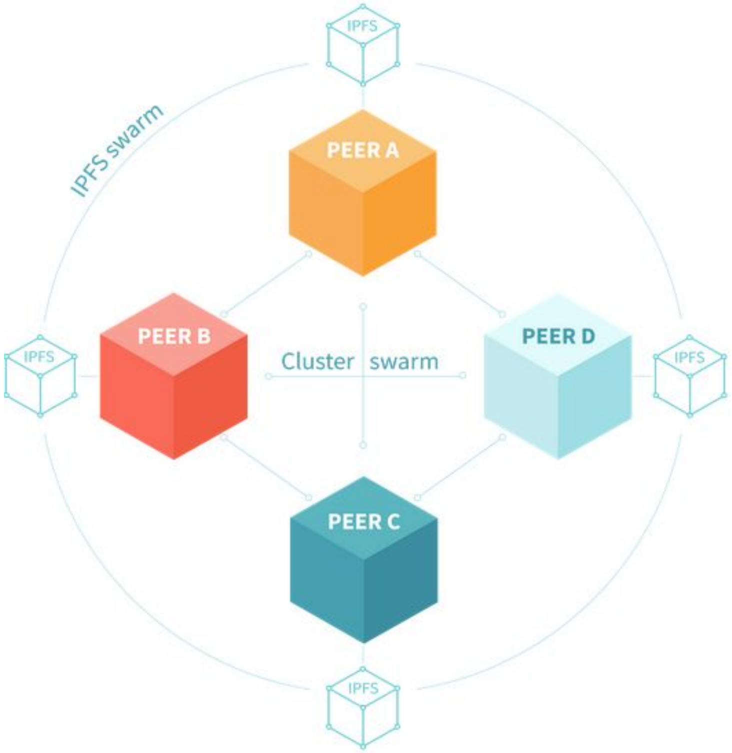
- It can be accessed using a simple curl command.
- Supports all the features and commands to interact with the node.
- Difficult to fetch file metadata from a distributed network.
- An example shown below.



```
C:\Users\Nirupam>curl -X POST http://52.0.65.207:5001/api/v0/id
{"ID":"QmZHrYMo73EDPt7k3254w7WmZP42EnJMVrGoSvcodQt5tu","PublicKey":"CAASpgIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDd3yzBg1BkP+BURF/j98V39ggq5W3IsbHLq
+i5oWPJ7+rTgURnvtE2RAgmVsdP1bByuFY+yMm8/SyPmzNkEO+T8HLlG/7/WhE1yRAEEU4QvztpfMLgsgftLvEpvDkoYKmOfQM1BsFmCmGf66764aLXOW73qTftmgCuxvlM5z9CCncmgURgVlrSnT8X1l
JwzAX4UA8ZUgnO+UAMw4oXlofXwGBR68CPgZSfLkRSewfwtT/BjhQlNAm0iQAUh5iSB38yyw7SGtf92RFCo+UPCOO503QKVwTXM/nZWfPyz20zp8eG+mVt+NQ6O0Y5lXRbutPtI5FV8ztV4UINu/yyTas
5AgMBAAE=","Addresses":["/ip4/127.0.0.1/tcp/4001/ipfs/QmZHrYMo73EDPt7k3254w7WmZP42EnJMVrGoSvcodQt5tu","/ip4/172.31.43.45/tcp/4001/ipfs/QmZHrYMo73EDPt7k32
54w7WmZP42EnJMVrGoSvcodQt5tu","/ip6/::1/tcp/4001/ipfs/QmZHrYMo73EDPt7k3254w7WmZP42EnJMVrGoSvcodQt5tu"],"AgentVersion":"go-ipfs/0.4.18/","ProtocolVersion"
:"ipfs/0.1.0"}
```

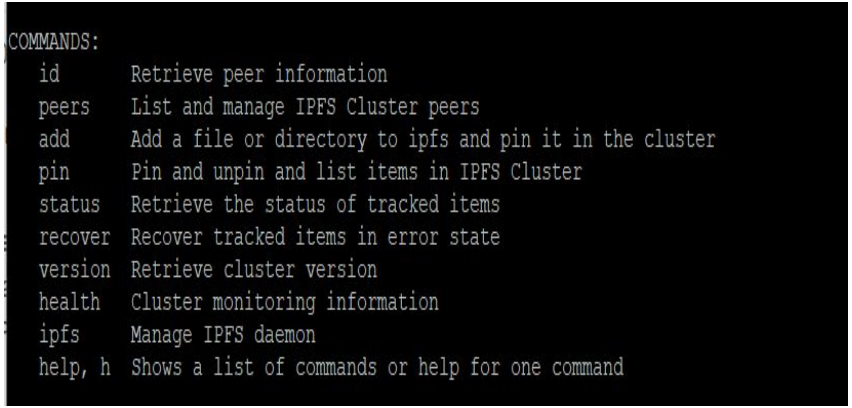
Approach - 2

- Using IPFS Cluster service
- Creating a private cluster of running IPFS nodes using the same secret key.
- Has a pinning service which saves and keeps track of files and peers.
- Data added to cluster is automatically and recursively pinned.



IPFS Cluster Service

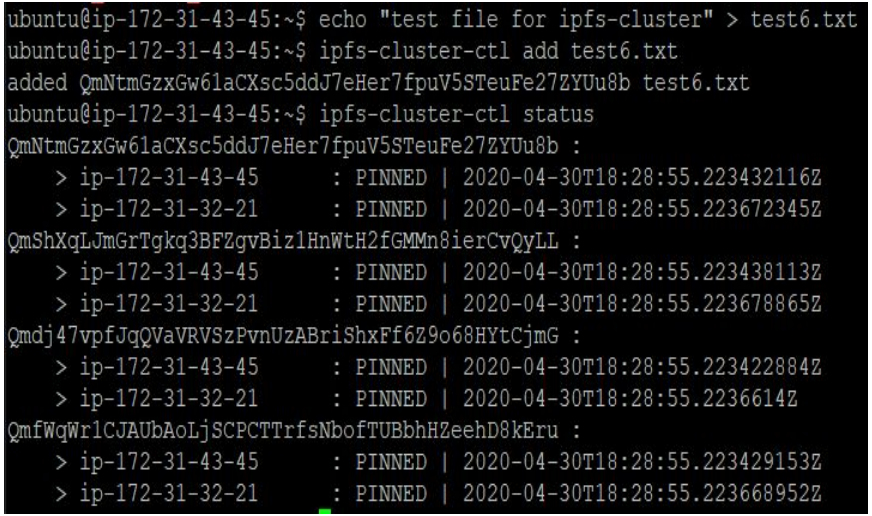
- Bootstraps IPFS nodes to form a cluster.
- Responsible for data replication across nodes.
- Updates changes in data across peers.
- Gives manual control for pinning data.
- Gives an interface to track files and check instances which have it pinned onto local storage.
- Provides many other diagnostic features for the network



```
COMMANDS:
   id        Retrieve peer information
   peers     List and manage IPFS Cluster peers
   add       Add a file or directory to ipfs and pin it in the cluster
   pin       Pin and unpin and list items in IPFS Cluster
   status    Retrieve the status of tracked items
   recover   Recover tracked items in error state
   version   Retrieve cluster version
   health    Cluster monitoring information
   ipfs      Manage IPFS daemon
   help, h   Shows a list of commands or help for one command
```

Cluster

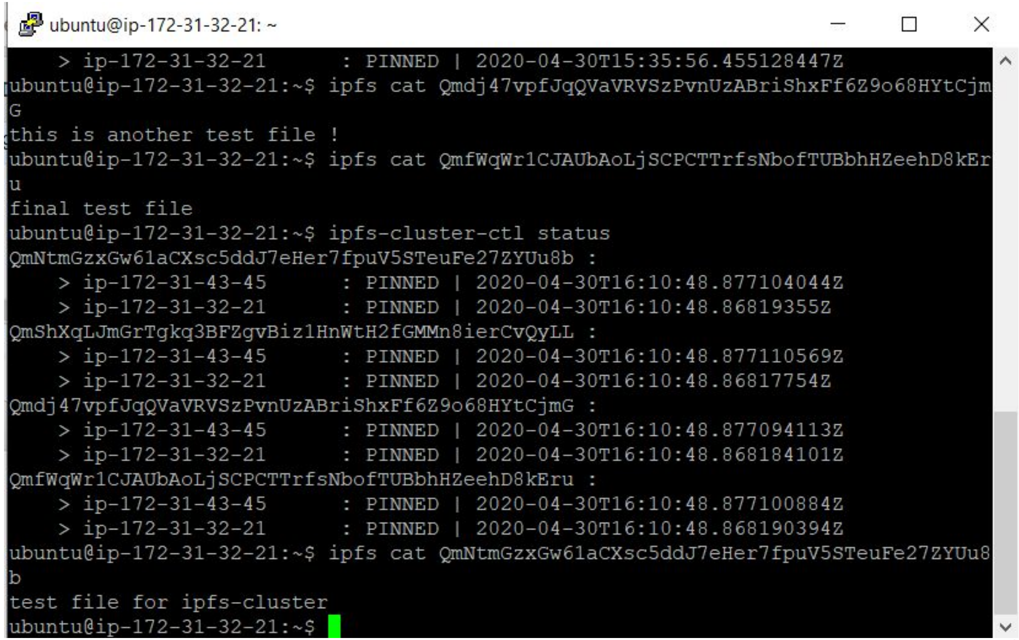
- Creating and adding a file to the cluster.
- Every file generates a unique hash and if the same file is uploaded it wont change the hash.
- We can see all the files in the network and which peers have it pinned in their storage.



```
ubuntu@ip-172-31-43-45:~$ echo "test file for ipfs-cluster" > test6.txt
ubuntu@ip-172-31-43-45:~$ ipfs-cluster-ctl add test6.txt
added QmNtmGzxGw61aCXsc5ddJ7eHer7fpuV5STeuFe27ZYUu8b test6.txt
ubuntu@ip-172-31-43-45:~$ ipfs-cluster-ctl status
QmNtmGzxGw61aCXsc5ddJ7eHer7fpuV5STeuFe27ZYUu8b :
    > ip-172-31-43-45       : PINNED | 2020-04-30T18:28:55.223432116Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T18:28:55.223672345Z
QmShXqLJmGrTgkq3BFZgvBiz1HnWtH2fGMMn8ierCvQyLL :
    > ip-172-31-43-45       : PINNED | 2020-04-30T18:28:55.223438113Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T18:28:55.223678865Z
Qmdj47vpfJqQVaVRVSzPvnUzABriShxFf6Z9o68HYtCjmG :
    > ip-172-31-43-45       : PINNED | 2020-04-30T18:28:55.223422884Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T18:28:55.2236614Z
QmfWqWr1CJAUbAoLjSCPCTTrfsNbofTUBbhHZeehD8kEru :
    > ip-172-31-43-45       : PINNED | 2020-04-30T18:28:55.223429153Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T18:28:55.223668952Z
```

Pinning

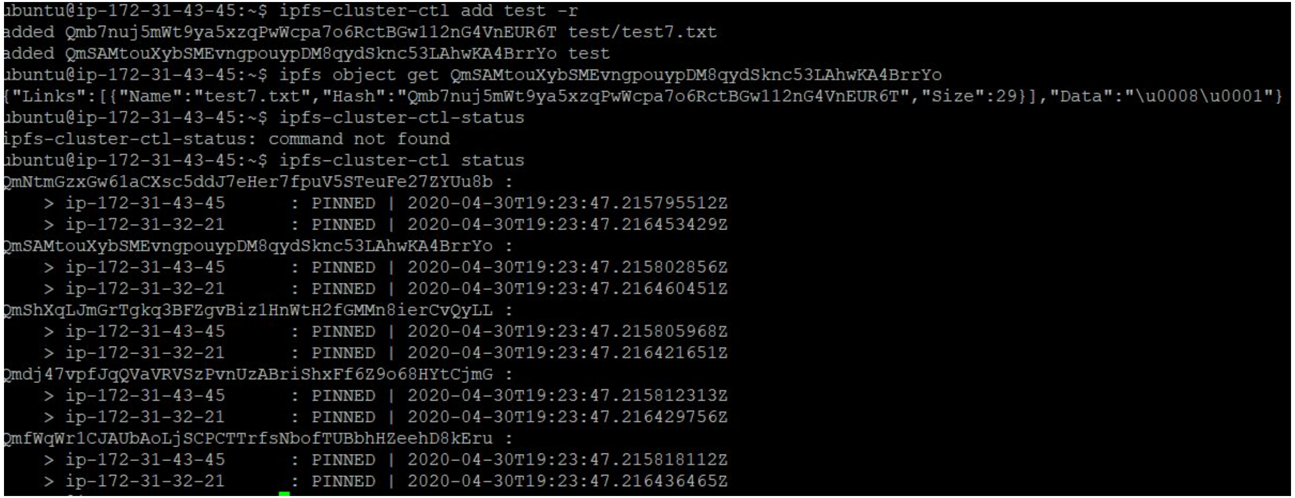
- Pinning is the mechanism that allows you to tell ipfs to always keep a given object in local storage.
- We can access files in the cluster using cat or get.
- IPFS has their own implementation of a unix like file system (mutable file system).
- IPFS Follows content addressing - It means that the file is represented by it's contents and not just it's name.



```
    > ip-172-31-32-21       : PINNED | 2020-04-30T15:35:56.455128447Z
ubuntu@ip-172-31-32-21:~$ ipfs cat Qmdj47vpfJqQVaVRVSzPvnUzABriShxFf6Z9o68HYtCjm
G
this is another test file !
ubuntu@ip-172-31-32-21:~$ ipfs cat QmfWqWr1CJAUbAoLjSCPCTTrfsNbofTUBbhHZeehD8kEr
u
final test file
ubuntu@ip-172-31-32-21:~$ ipfs-cluster-ctl status
QmNtmGzxGw61aCXsc5ddJ7eHer7fpuV5STeuFe27ZYUu8b :
    > ip-172-31-43-45       : PINNED | 2020-04-30T16:10:48.877104044Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T16:10:48.86819355Z
QmShXqLJmGrTgkq3BFZgvBiz1HnWtH2fGMMn8ierCvQyLL :
    > ip-172-31-43-45       : PINNED | 2020-04-30T16:10:48.877110569Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T16:10:48.86817754Z
Qmdj47vpfJqQVaVRVSzPvnUzABriShxFf6Z9o68HYtCjmG :
    > ip-172-31-43-45       : PINNED | 2020-04-30T16:10:48.877094113Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T16:10:48.868184101Z
QmfWqWr1CJAUbAoLjSCPCTTrfsNbofTUBbhHZeehD8kEru :
    > ip-172-31-43-45       : PINNED | 2020-04-30T16:10:48.877100884Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T16:10:48.868190394Z
ubuntu@ip-172-31-32-21:~$ ipfs cat QmNtmGzxGw61aCXsc5ddJ7eHer7fpuV5STeuFe27ZYUu8
b
test file for ipfs-cluster
ubuntu@ip-172-31-32-21:~$
```

File Metadata work around

- Everything in IPFS is a “block”.
- File metadata can be obtained by putting files in a folder and adding the folder to IPFS



```
ubuntu@ip-172-31-43-45:~$ ipfs-cluster-ctl add test -r
added Qmb7nuj5mWt9ya5xzqPwWcpa7o6RctBGw112nG4VnEUR6T test/test7.txt
added QmSAMtouXybSMEvngpouypDM8qydSknc53LAhwKA4BrrYo test
ubuntu@ip-172-31-43-45:~$ ipfs object get QmSAMtouXybSMEvngpouypDM8qydSknc53LAhwKA4BrrYo
{"Links":[{"Name":"test7.txt","Hash":"Qmb7nuj5mWt9ya5xzqPwWcpa7o6RctBGw112nG4VnEUR6T","Size":29}],"Data":"\u0008\u0001"}
ubuntu@ip-172-31-43-45:~$ ipfs-cluster-ctl-status
ipfs-cluster-ctl-status: command not found
ubuntu@ip-172-31-43-45:~$ ipfs-cluster-ctl status
QmNtmGzxGw61aCXsc5ddJ7eHer7fpuV5STeuFe27ZYUu8b :
    > ip-172-31-43-45       : PINNED | 2020-04-30T19:23:47.215795512Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T19:23:47.216453429Z
QmSAMtouXybSMEvngpouypDM8qydSknc53LAhwKA4BrrYo :
    > ip-172-31-43-45       : PINNED | 2020-04-30T19:23:47.215802856Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T19:23:47.216460451Z
QmShXqLJmGrTgkq3BFZgvBiz1HnWtH2fGMMn8ierCvQyLL :
    > ip-172-31-43-45       : PINNED | 2020-04-30T19:23:47.215805968Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T19:23:47.216421651Z
Qmdj47vpfJqQVaVRVSzPvnUzABriShxFf6Z9o68HYtCjmG :
    > ip-172-31-43-45       : PINNED | 2020-04-30T19:23:47.215812313Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T19:23:47.216429756Z
QmfWqWr1CJAUbAoLjSCPCTTrfsNbofTUBbhHZeehD8kEru :
    > ip-172-31-43-45       : PINNED | 2020-04-30T19:23:47.215818112Z
    > ip-172-31-32-21       : PINNED | 2020-04-30T19:23:47.216436465Z
```

Goals achieved

- Deployed an IPFS network on cloud using EC2 instances and VPC
- Creation of a private network with restricted access.
- File sharing among nodes.
- Configured and set up of the HTTP API
- Data Replication across all nodes.

Setbacks

- IPFSHTTP API is still in alpha and the features required are in development.
- IPFSHTTP API does not function properly with the cluster but works fine with individual nodes
- Browser Access/web interface was not possible due to the above reason.
- File metadata is accessible only when it is within a folder.
- Writing a GraphQL spec was difficult as the API was not structured as traditional REST APIs.

Future Prospects

- Implementation of a web interface
- Restful APIs in place of standard RPC APIs for node interaction.
- The ability to run cluster service and use the API at the same time
- Developing a high level library to simplify interactions with IPFS

References

- https://docs.ipfs.io/reference/api/http/
- https://www.npmjs.com/package/ipfs-cluster-api
- https://github.com/ipfs/ipfs-cluster
- https://en.wikipedia.org/wiki/InterPlanetary_File_System
- https://github.com/ipfs/go-ipfs

Questions?