

notes - 10/6/17

- what data is inside the directories
- recursively -ls
- peer-to-peer network
- 12 machines/beefy servers
- permissions - authentication for specific files
- on demand
- puts and post ?
- bitTorrent?
- farm hash - speed and minimal collisions
- in browser representation of the file system
- os.walk
- {status: success
 data: [f1, f2, f3]
 links: [/next, /nextto]}
- flask
- panda
- build docker container ***
- docker1 / docker2

spice/a/nextz

{ status: success

data : [f1, f2, f3]

links : [/next, /next] }

spice/a/nextz/f.jpg/raw
/mod

spice/a/nextz/cdirz/

~~The~~ Weekly @ 12

weird idea: everything in a pandas
data frame, hash to check for changes
Format hash: google, raw speed
min collision

Test 1: browser look at someone
else's files come back use
JSON responses
status: success
date: ~~empty~~ file1 file2...
links: ~~miss~~ /mission1 /mission2...
Just one level below

whoever has the latest is the
"master" for the file

Test 2: MD5 checksum
Test 3: client that can ask this

- Flask has testing for restful endpoints
- JSON can return a pandas DF
- Swagger can generate docs for an endpoint

one-way → look at other files
two-way → subscribe and ping update log
Stack: Flask/pandas

Docker website has tool for python proj.
Docker compose files: DI manager talks to DZ orchestrator
Possibly cache all changes →

Capstone 10/6/2017

Feffernoose
good for example
restful responses

What do we want:

- see what everyone has
- are they in sync
- peer to peer

(Not necessarily a ~~peer~~
distributed ~~peer~~)

-permissions

- some files private/public we want to be able to manage this and who gets to see (FLASK)

-use case: other guy adds 7 new files
get names, say you want to pull,
or check sum says diff names, figure out

STEP 1: Figure out one workstation's process

STEP 2: rest API talks into other
or it's stuff, MD5 checksums, get requests

STEP 3: post requests (give me)

We want some way to register
these together (peer-to-peer
connections to each other)

nice to have a restful in point
once they are together, know
to ping each other once an hour/day

→ out of sync, same name → MD5 checksum

→ out of sync, new name → restful