

# archomp

archive-compressor

orijtech

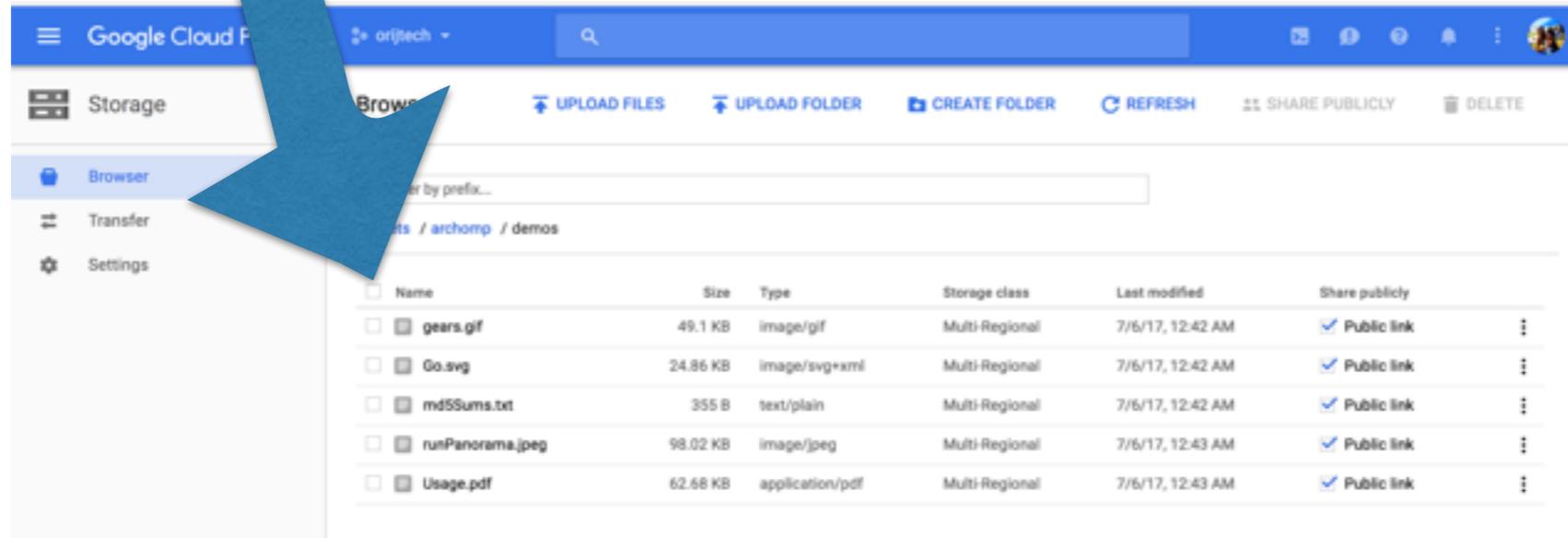
# archomp

- **Problem:** Many websites need to bundle resources, to archive, and zip content for their users to download instead of downloading one item at a time. Many websites may not have the resources to implement a backend to do downloading and archiving. The cost of running such a backend is a strain for entrepreneurs who want to build their idea without obstacles that need server expertise.
- **Solution:** I've implemented a language agnostic backend service that given a set of URLs, compresses them into a single zip that can be streamed or uploaded to your choice of destination. Orijtech is here to solve this problem, once and for all.

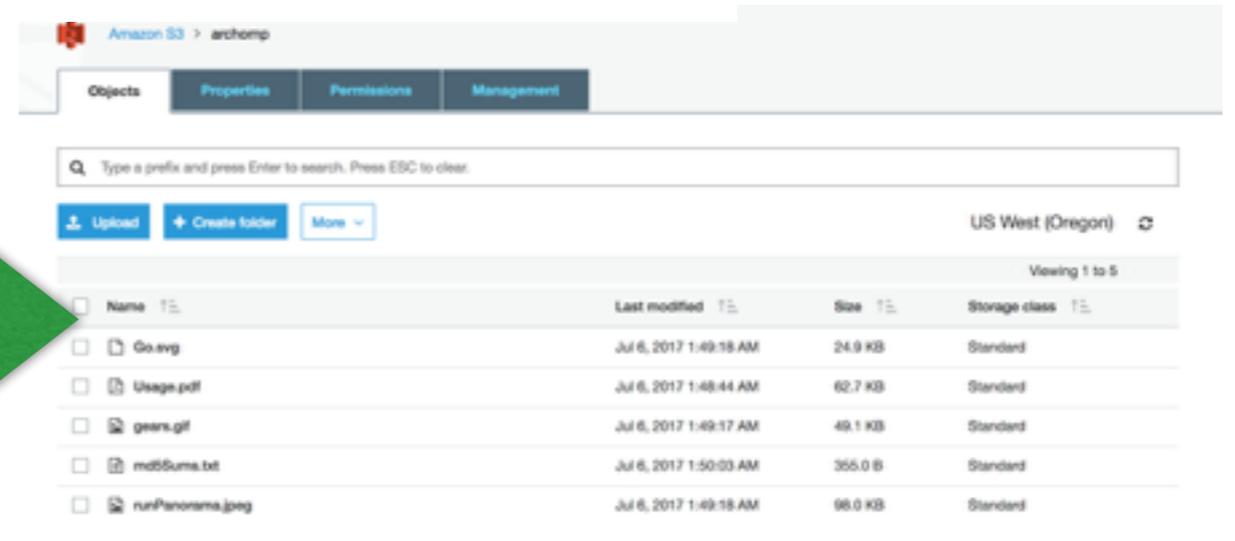
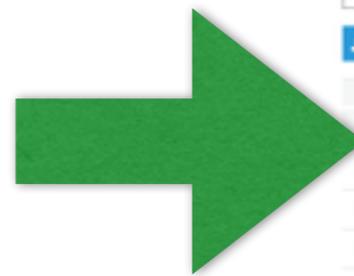
orijtech

# The problem illustrated in a few pictures

Currently no way to download multiple files at once on Google Cloud Storage



Same problem with Amazon S3

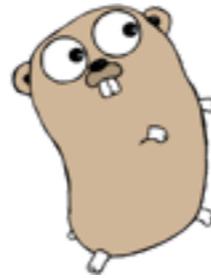


Add any service for you'll need to download multiple files as one, and the problem still exists unfortunately!

**orijtech**

# Solution in illustration: language agnostic API

```
archomp.go
1 package main
2
3 import (
4     "io"
5     "log"
6     "os"
7
8     "github.com/orijtech/orijgo/archomp/v1"
9 )
10
11 func main() {
12     f, err := os.Create("allCompressed.zip")
13     if err != nil {
14         log.Fatal(err)
15     }
16     defer f.Close()
17
18     client := new(archomp.Client)
19     rc, err := client.Compress(&archomp.Request{
20         Resources: []*archomp.Resource{
21             {URL: "https://storage.googleapis.com/archomp/demos/gears.gif"},
22             {URL: "https://storage.googleapis.com/archomp/demos/Go.svg"},
23             {URL: "https://storage.googleapis.com/archomp/demos/runPanorama.jpeg"},
24             {URL: "https://storage.googleapis.com/archomp/demos/Usage.pdf"},
25         },
26     })
27     if err != nil {
28         log.Fatal(err)
29     }
30     defer rc.Close()
31
32     io.Copy(f, rc)
33 }
```



```
archomp.py
1 import os, sys
2
3 import archomp.v1 as archomp
4
5 def main():
6     client = archomp.Client()
7     riter, err = client.compress([
8         {"url": "https://storage.googleapis.com/archomp/demos/gears.gif"},
9         {"url": "https://storage.googleapis.com/archomp/demos/Go.svg"},
10        {"url": "https://storage.googleapis.com/archomp/demos/runPanorama.jpeg"},
11        {"url": "https://storage.googleapis.com/archomp/demos/Usage.pdf"},
12    ])
13
14    if err is not None:
15        raise err
16
17    try:
18        zipname = 'zipped.zip'
19        with open(zipname, 'wb') as fzip:
20            for i, rchunk in enumerate(riter):
21                n = fzip.write(rchunk)
22                if n is None:
23                    continue
24
25                sys.stdout.write('chunk %d wrote %d bytes\r'%(i, n))
26
27            print('\nWrote %s to disk!\n'(zipname))
28
29    except Exception as ex:
30        raise ex
31
32    finally:
33        riter.close()
34
35
36 if __name__ == '__main__':
37     main()
```



```
p.sh
/bin/bash
curl -X POST https://archomp.orijtech.com/v1 --data \
'{"files": [{
  "url": "https://storage.googleapis.com/archomp/demos/gears.gif"
}, {
  "url": "https://storage.googleapis.com/archomp/demos/Go.svg"
}, {
  "url": "https://storage.googleapis.com/archomp/demos/md5Sums.txt"
}, {
  "url": "https://storage.googleapis.com/archomp/demos/runPanorama.jpeg"
}, {
  "url": "https://storage.googleapis.com/archomp/demos/Usage.pdf"
}]
}' > all.zip
```



```
archomp.js
1 const fs = require('fs');
2 const archomp = require('archomp');
3
4 function main() {
5     var arc = new archomp({apiKey: 'd8aa7829-e5d1-4d7f-8917-5327853f9bee'});
6
7     arc.compress({
8         "files": [{
9             "url": "https://orijtech.com/", "name": "orijtech-homepage.html"
10        }, {
11        "url": "https://tatan.s3-us-west-2.amazonaws.com/uno.gif", "name": "guap.gif"
12    }]
13    }, function(statusCode, r) {
14        if (statusCode !== 200) {
15            console.log(r);
16            return;
17        }
18
19        var outfile = 'outf.zip'
20        fs.writeFile(outfile, r, function(err) {
21            if (err)
22                throw err;
23
24            console.log('Successfully saved to %s!', outfile);
25        });
26    });
27 }
28
29 if [process.argv.length >= 2 && process.argv[1] !== __filename]
30     main();
```



you name the language, it can be ported to...

# orijtech

# archomp advantages

- Easy to add to any front-end, mobile, or backend system; libraries can be implemented in any language, due to the language agnosticity of the service and API specification
- archomp relieves your app from the pressures of scaling servers, and hiring expensive software maintainers. It should be simple for anyone that needs to dispense items in bulk to their customers for example:
  - logo vendors
  - digital libraries
  - cloud storage services
  - on boarding apps
  - academic-focused apps that need to give you schedules on the fly
  - academic institutions that need to bundle your transcripts with other records
  - online shopping stores that dispense multiple receipts
  - accountants and taxation agency to give your business or person an archive of all their receipts for a time period.
- Let your imagination run wild

orijtech

Questions?  
Comments?  
Thoughts?

# Thank you!

Emmanuel Odeke  
[emmanuel@orijtech.com](mailto:emmanuel@orijtech.com)