

# Seaweedfs Distributed Storage Part 2: Reading and Writing Files' Process.

Ali Hussein Safar · Follow  
3 min read · Sep 8

4

In Seaweedfs, two different approaches can be followed to read or write files to the cluster:

## A- Writing and Reading Files to Seaweedfs without using the Filer Service

To write a file, the following steps need to be followed:

1. client needs to ask the master server to know which volume server can be used to store the file as shown in the following API request to the master server.

```
curl http://MasterIP:9333/dir/assign
Response: {"fid":"7,016bad1bc0","url":"VolumeServerIP:8080"}
```

2. Then client can upload the file using the following request with the file to this volume server

```
Curl -F file=@/file.txt http://VolumeServerIP:8080/7,016bad1bc0
```

1. Once the file is uploaded successfully, the client needs to store the uploaded file name, volume ID, and file ID (In our case it would be file.txt, 7, 016bad1bc0) in the database so that it could be available for future read requests.

The following steps can be used to access the stored file in the cluster.

1. Client needs to query the database to fetch volume ID, file ID for the requested file. In our example, the returned data for file.txt should be (7,016bad1bc0) where 7 is volume ID and 016bad1bc0 is file ID.
2. Client will need to send a get request to the master node to know in which volume server volume 7 exists because volumes might be migrated between volume servers.

```
curl http://MasterIP:9333/dir/lookup?volumeId=7
Response: {"volumeId":"7","locations":[{"url":"VolumeServerIP:8080"}]}
```

1. Then a final get request to the volume to fetch the file

```
curl http://VolumeServerIP:8080/7,016bad1bc0
```

To ease the process of storing and retrieving files in the cluster seaweed team has created two services which are file and filer store.

## B- Writing and Reading Files to Seaweedfs using the Filer Service

As mentioned before the filer service provides different entry points to the cluster one of them is the filer Web UI which allows us to download or upload files through the web. The following describes the process of writing a new file

1. The client uploads a file to the filer using the Web UI.
2. The filer will ask the master to provide the volume server IP, volume ID, and file ID to store the file in the volume server. However, according to filer configuration, files might be separated into chunks and stored as separate files in the volume servers to increase read and write speed.
3. The filer will store returned data from the master in the filer store and sync the other filer nodes in case an embedded filer approach is used.
4. The filer will upload the file to the volume server.

In case of accessing the file from the filer Web UI:

1. From the Web UI, the client will press on one of the files to download it.
2. The filer will ask the filer store to provide Volume ID, file ID of the requested file.
3. The filer will pull the file from the volume servers and provide it to the user. However, if the file is chunked then the filer will pull all the chunks and then combine the chunks to form the file, and then return the requested file to the user.

[Part 1: Introduction.](#)

[Part 3: Features.](#)

### Resources

- <https://github.com/seaweedfs/seaweedfs/wiki>.
- [https://www.usenix.org/legacy/event/osdi10/tech/full\\_papers/Beaver.pdf](https://www.usenix.org/legacy/event/osdi10/tech/full_papers/Beaver.pdf).

Thank you for reading my article on SeaweedFS. I hope you find it informative and helpful. If you enjoy the article and would like to support my work, follow me or you can buy me a coffee at <https://www.buymeacoffee.com/ahsifer>. Your support is greatly appreciated

Storage Distributed Storage Object Storage Linux System Administration

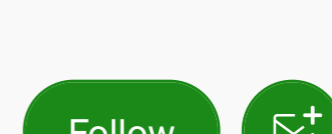
4



Written by Ali Hussein Safar

11 Followers

System administrator



### More from Ali Hussein Safar

Ali Hussein Safar

**Seaweedfs Distributed Storage Part 3: Features.**

SeaweedFS supports a wide range of interesting features:

5 min read · Sep 8

9

Ali Hussein Safar

**SeaweedFS Containerized Cluster Deployment with Fuse Mount...**

SeaweedFS is a distributed storage system for blobs, objects, and files with predictable...

10 min read · Oct 28

13

Ali Hussein Safar

**Seaweedfs Distributed Storage Part 1: Introduction.**

SeaweedFS is a distributed storage system for blobs, objects, files with predictable low...

5 min read · Sep 5

5

See all from Ali Hussein Safar

### Recommended from Medium

Vagharsh Kandilian

**Push multi-container Kubernetes pod logs to AWS CloudWatch usin...**

I'm not going to delve into the history of why we need to use the AWS Elastic Kubernetes...

7 min read · Jul 9

9

Ali Hussein Safar

**SeaweedFS Distributed Storage Part 3: Features.**

SeaweedFS supports a wide range of interesting features:

5 min read · Sep 8

9

### Lists

**General Coding Knowledge**

20 stories · 653 saves

**Staff Picks**

530 stories · 499 saves

Hakan Bayraktar

**How to Install Kubernetes Cluster on Ubuntu 22.04 (Step-by-Step...**

Introduction

6 min read · Nov 2

101

BigDataEnthusiast

**MinIO—High Performance Object Storage**

MinIO is a high-performance, kubernetes native object storage.

4 min read · Aug 20

2

Santhosh Kumar BVSRK

**Oracle Autonomous Database : Connect using Wallet & No-Wallet**

Use-case

5 min read · Oct 6

1

David Goudet

**This is Why I Didn't Accept You as a Senior Software Engineer**

An Alarming Trend in The Software Industry

5 min read · Jul 26

5.1K

See more recommendations