



Web



Configuration



WEB &gt; GUIDES &gt; PSPDFKIT SERVER &gt; CONFIGURATION

# Optimize asset storage for better PDF management



PSPDFKit Server has been deprecated and replaced by [Document Engine](#). To migrate to Document Engine and unlock advanced document processing capabilities, refer to our [migration guide](#). Learn more about these enhancements on our [blog](#).

PSPDFKit Server supports multiple storage backends for PDFs and other assets, as detailed below.

## Built-in asset storage

By default, PSPDFKit Server stores assets as Binary Large Objects (BLOBs) in the database. If you have individual PDFs that are bigger than 1 GB in size, we recommend using [S3-compatible object storage](#).

Set `ASSET_STORAGE_BACKEND` to `built-in` to use the built-in asset storage.

## S3-compatible object storage

PSPDFKit Server can also store your assets in any [Amazon S3-compatible](#) object storage service.

### Configuration



Set `ASSET_STORAGE_BACKEND` to `s3`. Other configuration options depend on whether you're using AWS S3 or another S3-compatible storage provider.

## AWS S3

When running on S3, you must set the `ASSET_STORAGE_S3_BUCKET` and `ASSET_STORAGE_S3_REGION` configuration options to configure the bucket name and region.

If you're running on AWS, Server will try to resolve access credentials with the following precedence:

- ❖ `ASSET_STORAGE_S3_ACCESS_KEY_ID` and `ASSET_STORAGE_S3_SECRET_ACCESS_KEY` configuration options
- ❖ [ECS Task Role](#)
- ❖ [EC2 Instance Role](#)

We recommend using ECS Task Role or EC2 Instance Role, as they don't require you to distribute credentials to the Server container via environment variables.

If you're not running on AWS, you must always set `ASSET_STORAGE_S3_ACCESS_KEY_ID` and `ASSET_STORAGE_S3_SECRET_ACCESS_KEY`.

## Other S3-compatible storage providers

When using an object storage provider other than [Amazon S3](#), you must always set `ASSET_STORAGE_S3_ACCESS_KEY_ID` and `ASSET_STORAGE_S3_SECRET_ACCESS_KEY`. In addition, you can configure the following options:

- ❖ `ASSET_STORAGE_S3_HOST` — Host name of the storage service.
- ❖ `ASSET_STORAGE_S3_PORT` — Port used to access the storage service. The default port is `443`.
- ❖ `ASSET_STORAGE_S3_SCHEME` — A URL scheme used when accessing the service, either `http://` or `https://`. The default is `https://`.

For more details about using [Google Cloud Storage](#) as the storage backend, take a look at the Google Cloud Storage interoperability guide.

## Bucket policy

If you're using AWS S3, the IAM identity used by PSPDFKit Server needs the following permissions:

- ❖ `s3:ListBucket` on the configured bucket

- ❖ `s3:PutObject` on all objects in the bucket ( `<bucket-arn>/*` )
- ❖ `s3:GetObjectAcl` on all objects in the bucket ( `<bucket-arn>/*` )
- ❖ `s3:GetObject` on all objects in the bucket ( `<bucket-arn>/*` )
- ❖ `s3:DeleteObject` on all objects in the bucket ( `<bucket-arn>/*` )

## Timeouts

Note that all operations on the S3 bucket have a timeout of 30 seconds.

## Which storage backend should I use?

The choice of storage backend depends on the PDF dataset that will power your application, and it impacts the general performance of PSPDFKit Server.

If you have a relatively stable number of PDF files (i.e. an amount that only changes a few times a month) with a size of lower than 5 MB each, you can safely use the built-in storage, with the main advantages being that:

- ❖ You don't have to worry about another piece of infrastructure.
- ❖ Backing up the PSPDFKit Server PostgreSQL instance will also back up your assets.

For larger and more frequently changing files, we recommend using the S3-compatible asset storage backend, which provides more efficient support for concurrent uploads and downloads.

Using the S3-compatible backend means you need a separate backup routine, but you should consider that:

- ❖ As PSPDFKit Server stores files by their SHA checksums, most of the time, a daily, incremental backup will suffice.
- ❖ You should schedule the asset storage backup right after the PostgreSQL database backup to avoid data drifting between the two.

## Serving files from existing storage in your infrastructure

If you already have a storage solution for PDF files in your infrastructure, PSPDFKit Server can integrate with it as long as the PDF files can be accessed via an HTTP endpoint. When integrating PSPDFKit

Server and the file storage, you'll need to [add documents from a URL](#).

All PDF URLs should be considered permalinks, as PSPDFKit will always fetch the file when needed (keeping only a local cached copy that can expire at any time).



Never accept arbitrary user input as a URL for a PDF. Malicious users might leverage this to make the server perform a request on their behalf. This kind of attack, known as Server-Side Request Forgery (SSRF), can be used to interact with services that assume the local network is secure, e.g. cloud automation infrastructure.

To achieve the best possible performance, ensure PSPDFKit Server instances and the file store sit in the same network (physical or virtual). This minimizes latency and maximizes download speed.

As of version 2019.4, it's possible to perform a document editing operation on a document with a remote URL, but the resulting PDF file will need to be stored with any of the supported storage strategies. If you need to copy the transformed file back to the file store, you'll need to do that manually by [fetching the transformed file first](#).

If your file store requires authentication, we recommend introducing an internal proxy. When adding a document with a URL, the URL would point to the proxy endpoint, where your custom logic would be able to support the required authentication options and redirect to the file store URL of the PDF file. For more information and some sample code, visit the [relevant guide article](#).

## MinIO

Our recommended solution when using an S3-compatible object storage in production is to use [MinIO](#) in development, in order to get closer to dev/prod parity.

To run the MinIO Docker container, run the following:

```
1 docker pull minio/minio
2 docker run -p 9000:9000 minio/minio server /export
```



After running these commands, you should see the `AccessKey` and `SecretKey` printed out in the terminal, which you can use to access the MinIO web interface at `http://localhost:9000/minio`.

You can now configure `docker-compose.yml`, like this:



```
1  environment:
2  ASSET_STORAGE_BACKEND: S3
3  ASSET_STORAGE_S3_BUCKET: <minio bucket name>
4  ASSET_STORAGE_S3_ACCESS_KEY_ID: <minio access key>
5  ASSET_STORAGE_S3_SECRET_ACCESS_KEY: <minio secret access key>
6  ASSET_STORAGE_S3_SCHEME: http://
7  ASSET_STORAGE_S3_HOST: pssync_minio
8  ASSET_STORAGE_S3_PORT: 9000
```

MinIO supports emulating different regions. It defaults to `us-east-1`. If you've changed your MinIO configuration to a different region, make sure to set `ASSET_STORAGE_S3_REGION` accordingly.

## Migration between asset storage options

It's possible to migrate from one storage backend to another one by executing the migration command as described below. To prevent data loss, a migration doesn't delete files from the original storage backend.

Asset storage backend migrations are incremental. You can interrupt the migration process at any time and resume it later on. This is useful when you have many documents and you'd like to perform the migration only during the time of low load on your system. You can perform the migration while PSPDFKit Server is running.

Before you start the migration process, make sure to set the `ENABLE_ASSET_STORAGE_FALLBACK` configuration option to `true`. This will allow PSPDFKit Server to serve assets that haven't yet been migrated from the old storage backend. Remember to set it back to `false` when you've finished migrating all the documents, as it introduces a slight decrease in performance of fetching the assets.

At any point, you can inspect how many documents are stored in each asset storage backend from the Storage tab in the [PSPDFKit Server dashboard](#).

## Migrating to S3 from built-in storage

To migrate from the built-in asset storage to S3, follow these steps:

- 1 Set the `ENABLE_ASSET_STORAGE_FALLBACK` configuration option to `true`.
- 2 Set the `ASSET_STORAGE_BACKEND` configuration option to `s3` and configure the rest of the S3 options.
- 3 Run the migration script by executing the `pspdfkit assets:migrate:from-built-in-to-s3` command in the PSPDFKit Server container.

- ⚙️ If you use docker-compose, run the following command in the directory where you have your `docker-compose.yml` file: `docker-compose run pspdfkit pspdfkit assets:migrate:from-built-in-to-s3`.
  - ⚙️ If you don't use docker-compose, first find the name of the PSPDFKit Server container using `docker ps -a`. This will list all running containers and their names. Then, run the following command, replacing `<container name>` with the actual PSPDFKit Server container name:  
`docker exec <container name> pspdfkit assets:migrate:from-built-in-to-s3`.
- 4 When all your documents have been migrated, set the `ENABLE_ASSET_STORAGE_FALLBACK` option back to `false`.

## Migrating to built-in storage from S3

To migrate from the S3 asset storage to the built-in storage, follow these steps:

- 1 Set the `ENABLE_ASSET_STORAGE_FALLBACK` configuration option to `true`.
  - 2 Set the `ASSET_STORAGE_BACKEND` configuration option to `built-in`. Do not remove any of the S3 configuration options.
  - 3 Run the migration script by executing the `pspdfkit assets:migrate:from-s3-to-built-in` command in the PSPDFKit Server container.
- ⚙️ If you use docker-compose, run the following command in the directory where you have your `docker-compose.yml` file: `docker-compose run pspdfkit pspdfkit assets:migrate:from-s3-to-built-in`.
  - ⚙️ If you don't use docker-compose, first find the name of the PSPDFKit Server container using `docker ps -a`. This will list all running containers and their names. Then, run the following command, replacing `<container name>` with the actual PSPDFKit Server container name:  
`docker exec <container name> pspdfkit assets:migrate:from-s3-to-built-in`.
- 4 When all your documents have been migrated, set the `ENABLE_ASSET_STORAGE_FALLBACK` option back to `false` and remove all the S3 configuration options.
- 

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

