

Connect into an external storage on S3 and register items in V7

This guide will follow you through steps needed for connecting into your external storage on AWS S3 first, and then registering new items with V7 Darwin.

[warning-note] The external storage integrations are only available on V7's Business and Enterprise plans. You can find out more about what each plan includes on our [pricing page](#).

Steps Overview

Here are the steps described in this guide from a high-level perspective:

1. Give your AWS role a Read and Write or Read-only [access](#), according to your needs
2. Configure a [CORS header](#) from within the AWS S3 Console
3. Request external storage [activation](#) from V7 support team
4. Register new items with V7 using specific [Read and Write](#) or [Read-only](#) python scripts

Connect to AWS

Read/Write Access

To setup an external S3 account, you will first need to give your AWS role (AWS key in the code sample below) the following access:

- [required] *Read* via GetObject
- [optional] *Write* via PutObject – needed for Darwin to be able to process data after it's uploaded (e.g. generate thumbnails, split video frames, etc.)

[info-note] Make sure that the policies are added directly to your S3 bucket (resource-based policy) not just via IAM (role-based policy). In case of doubt we recommend following [this AWS guide](#).

[1st tab - Read and Write]

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForExternalAccess",
  "Statement": [
    {
      "Sid": "DarwinAccess",
```

```

    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::258327614892:role/external_s3"
    },
    "Action": ["s3:GetObject", "s3:PutObject"],
    "Resource": "arn:aws:s3:::your-s3-bucket-name/*"
  }
]
}

```

[2nd tab - Read-only]

```

{
  "Version": "2012-10-17",
  "Id": "PolicyForExternalAccess",
  "Statement": [
    {
      "Sid": "DarwinAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::258327614892:role/external_s3"
      },
      "Action": ["s3:GetObject"],
      "Resource": "arn:aws:s3:::your-s3-bucket-name/*"
    }
  ]
}

```

CORS Access

When annotators are requesting images to annotate, they will be able to load them directly from your S3 bucket via a presigned URL.

However, since that S3 bucket sits on a different domain than `darwin.v7labs.com`, a CORS header needs to be configured first.

You can find this under **Permissions > CORS Configuration** in the [AWS S3 Console](#):

```

[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [

```

```
        "GET"  
    ],  
    "AllowedOrigins": [  
        "https://darwin.v7labs.com"  
    ],  
    "ExposeHeaders": []  
  }  
]
```

Activation

Once you have setup appropriate access according to the steps above, send an email to support@v7labs.com with the following details:

- S3 region
- S3 bucket name
- an optional prefix where we can upload thumbnails if needed (often `/darwin/`)
- your team name

and we will turn on the external access for your team.

[warning-note] Note that bucket names containing dots (.) will not work due to how AWS handles virtual-host-style HTTPS. Read more about it in [AWS documentation](#).

Register New Images or Videos

After the external bucket is ready and V7 support team has enabled access, you need to notify Darwin which images or videos to list for which dataset. This is done via a REST PUT request.

[warning-note] All REST endpoints in Darwin are using API keys for authentication. Before moving to the next step, check if you have your own API key set up [here](#).

Read and Write Access

If the external storage configurations allow V7 write access, the frames and thumbnails can be extracted by V7 automatically after registering the items.

The following python scripts show how to register a newly added image or video:

[\[1st tab - Images\]](#)

```

import requests

api_key = "<API_KEY HERE>"

team_slug = "your-team-slug-here"
dataset_slug = "your-dataset-slug-here"
storage_name = "your-storage-bucket-name-here"

headers = {
    "Content-Type": "application/json",
    "Accept": "application/json",
    "Authorization": f"ApiKey {api_key}"
}

payload = {
    "items": [
        {
            "path": "/",
            "slots": [
                {
                    "as_frames": "false",
                    "slot_name": "1",
                    "storage_key": "car_folder/cars.jpg",
                    "file_name": "cars.jpg"
                }
            ],
            "name": "cars.jpg"
        }
    ],
    "dataset_slug": dataset_slug,
    "storage_slug": storage_name
}

response = requests.post(
    f"https://darwin.v7labs.com/api/v2/teams/{team_slug}/items/register_existing",
    headers=headers,
    json=payload
)

if response.status_code != 200:
    print("request failed", response.text)
else:
    print("success")

```

[2nd tab - Videos]

```
import requests

api_key = "<API_KEY HERE>"

team_slug = "your-team-slug-here"
dataset_slug = "your-dataset-slug-here"
storage_name = "your-storage-bucket-name-here"

headers = {
    "Content-Type": "application/json",
    "Accept": "application/json",
    "Authorization": f"ApiKey {api_key}"
}

payload = {
    "items": [
        {
            "path": "/",
            "slots": [
                {
                    "as_frames": False,
                    "slot_name": "1",
                    "fps": 10,
                    "file_name": "seals.mp4",
                    "storage_key": "seals_folder/seals.mp4",
                    "type": "video"
                }
            ],
            "name": "Seals"
        }
    ],
    "dataset_slug": dataset_slug,
    "storage_slug": storage_name
}

response = requests.post(
    f"https://darwin.v7labs.com/api/v2/teams/{team_slug}/items/register_existing",
    headers=headers,
    json=payload
)

if response.status_code != 200:
```

```
print("request failed", response.text)
else:
    print("success")
```

Read-only Access

In case you prefer to generate thumbnails on your own, or when you are unable to give write access to V7 to your bucket, the `payload` changes slightly.

In code samples below, you can see that `thumbnail_key`, `height` and `width` keys are added for images.

For videos, `thumbnail_key`, `hq_key` (high quality frames used for annotation when the video is paused) and `lq_key` (low quality frames used during playback), as well as the `height` and `width` for each frame, are added.

[danger-note] We recommend not to use the originally registered image for your thumbnail. This can lead to CORS issues on some browsers.

[1st tab - Images with own Thumbnails]

```
"items": [
  {
    "path": "/",
    "slots": [
      {
        "type": "image",
        "storage_key": "darwin/cars/2008_000074.jpg",
        "storage_thumbnail_key":
"darwin/2008_000074_thumbnail.jpg",
        "width": 1920,
        "height": 1080,
        "file_name": "2008_000074.jpg"
      }
    ],
    "name": "cars.jpg"
  }
]
```

[2nd tab - Video with own Thumbnails]

```
payload = {
  "items": [
    {
```

```

    "path": "/",
    "as_frames": False,
    "fps": 2,
    "sections": [
        {
            "height": 1200,
            "section_index": 1,
            "width": 1400,
            "storage_hq_key": "hq_key_1.jpg",
            "storage_lq_key": "lq_key_1.jpg"
        },
        {
            "height": 1200,
            "width": 1400,
            "section_index": 2,
            "storage_hq_key": "hq_key_2.jpg",
            "storage_lq_key": "lq_key_2.jpg"
        }
    ],
    "name": "my-video.mp4",
    "storage_key": "storage_path.jpg",
    "storage_thumbnail_key": "thumbnail_key.jpg",
    "type": "video"
},
{
    "dataset_slug": "my-dataset",
    "storage_slug": "my-storage-bucket"
}
}

```

For more details, tips or key definitions – please visit our detailed [Item Registration documentation](#).

What's Next?

- If you are interested in registering DICOM or NIFTI files, see our [dedicated documentation](#) on working with slots.
- Visit V7 [API Reference](#) for more details on managing external storage settings.

Need help?

If you encounter any issues or have any additional questions, please contact our support team at support@v7labs.com.